Programming /Verification Specification

Date: Mar 24, 2014
Document Release: Version 2.30
Devices Covered: LPC1111/1112/11U12/1113/1114
LPC1101LV/1102LV/1112LV/1114LV
LPC11A02/11A04/11A11/11A12/11A13/11A14
LPC11C12/11C14/11C22/11C24
LPC11D14
LPC11E11/11E12/11E13/11E14
LPC11U12/11U13/11U14/11U23/11U24/11U34/11U35/11U36/11U37
LPC1224/1225/1226/1227/12D27
LPC12D27
LPC1311/13/15/16/17/42/43/45/46/47
LPC1751/52/54/56/58/59/64/65/66/67/68/69/74/76/77/78/85/86/87/88
LPC1812/13/15/17/22/23/25/27/33/37/53/57
LPC2101/2/3
LPC2104/2105/2106
LPC2114/2124/2124/2109/2119/2129/2194
LPC2131/2132/2134/2136/2138
LPC2141/2142/2144/2146/2148
LPC2212/2214/2292/2294
LPC2361/2362/2364/2365/2366/2367/2368/2377/2378/2387/2388
LPC2458/2468/2478
LPC4072/4074/4076/4078/4088
LPC4312/13/15/17/22/23/25/27/33/37/53/57

Boot Code Parallel:
2.0 for LPC21xx/22xx only
3.0 for LPC2361/62/64/65/66/67/68/77/78/87/88/2458/68/78 only
4.0 for LPC1751/52/54/56/58/63/64/65/66/67/68 only
5.0 for LPC1311/13/42/43 only
6.0 for LPC1111/12/C12/13/14/C14/D14 only
6.1 for LPC11U12/U13/U14/U23/U24/E11/E12/E13/E14 and LPC1315/16/17/45/46/47 only
6.2 for LPC11A02/11A04/11A11/11A12/11A13/11A14 only
6.3 for LPC1111FD20/1112FD20/1112FDH20, LPC11xxLV
7.0 for LPC1112FHN24
8.0 for LPC1224/25/26/27/D27 only
9.0 for LPC1774/76/77/78/85/86/87/88 and LPC4072/74/76/78/88 only
10.0 for LPC181x/2x/3x/5x and LPC431x/2x/3x/5x

REVISION HISTORY

V1.0

- Initial document release

V1.1

- Typo errors in Table 1 corrected (pin P0.0 and P0.1 function during programming)
- Paralell programming details added
- Table 24 containing required messages during the programming process updated

V1.2

- Table 24 containing required messages during the programming process updated

V1.3

- Table 24 containing required messages during the programming process updated
- Scope of the programming specification extended to the LPC2112, LPC2114, LPC2124, LPC2124, LPC2119, LPC2129, LPC2194, LPCLPC2212, LPC2214, LPC2292 and LPC2294
- Parallel load of the interface code replaced with the serial load of 1K of UUENCODED data; parallel programming is therefore no more on-chip bootloader version dependent
- Requirement for external pull-ups on pins P0.2 through P0.6 is added into the Table 1

V1.4

- Flash memory code read protection feature added to the programming specification
- LPC2131, LPC2132, and LPC2138 added to the specification
- Details on using external crystals higher than 10 MHz added to the document

V1.41

- Order of bytes in Flash memory that enable Code Read Protection (CRP) corrected

V1.5

- LPC2134 and LPC2136 added to the specification
- A remark on the CRP effectiveness after the power-up added
- Note 5 in Table 1 updated

V1.6

- LPC2141, LPC2142, LPC2144, LPC2146, and LPC2148 added to the specification

V1.7

- LPC2103 added to the specification
- A note no.6 is added in Table 1 requering an external pull-up on pins P0.11 and P0.14 when programming the LPC2131/2/4/6/8 and LPC2141/2/4/6/8 devices

V1.71

- LPC2101/2 devices are added to the spec together with details on LPC2101/2/3 Vbat pin conditioning during programming

V1.8

- Details on making a code self executable after a reset (0x0000_0014 vector handling) added to the spec
- LPC2364/2366/2368/2378 and LPC2468 as well as dedicated Boot Code Parallel 3.0 are added to the specification

V1.81

- LPC2109 device added to the spec

V1.82

- Typographic errors in Table 25 (message number 5 and message number 6 notes) corrected

V1.83

- Table 25 updated with sector 27 for LPC2368/2378/2468 devices; full chip erase implementation depends on this

V1.84

- LPC2365/2367/2377/2387/2388/2458/2478 devices added to the spec
- LPC2364/2368/2468 updated with TFBGA package details
- Table 2 and Table 3 updated with NC (no connect) requirements
- Table 37 updated with new PartIDs
- Table 46 updated with added devices' bootloader versions

V1.85

- LPC2361/2362 devices added to the spec

V1.86

LPC2387 Part ID information updated

V1.87

- LPC2292 updated with TFBGA package details
- Example section figure references updated

V2.00

- Support for LPC1751/52/54/56/58/64/65/66/67/68 added (including a dedicated Boot Code Parallel 4.0)

V2.01

- Support for LPC1311/13/42/43 added (including a dedicated Boot Code Parallel 5.0)

V2.02

- Support for LPC1759/69 added

V2.03

- Support for LPC1111/12/13/14 added
- Some typographic errors corrected
- Pin multiplexing notes added

V2.04

- Support for LPC11C12/11C14 added

V2.05

- Support for LPC1763 added
- typos in LPC11/13/1700 Flash Memory Sectors and Addresses accessible for parallel programming tables fixed

V2.06

- Flash Memory Code Read protection chapter updated

V2.07

- PartID information updated with the latest LPC11xx device details

V2.08

- Support for LPC1224/5/6/7 added
- LPC23xx/24xx V_{DD3x}/V_{DD3} pin list updated (see Table 2 and Table 3)

V2.09

- Supported LPC1224/5/6/7 PartIDs updated

V2.10

- Support for LPC1311/01 and LPC1313/01 added

V2.11

- Support for LPC1774/76/77/78/85/86/87/88 added
- Support for LPC11U12/U13/U14 added
- Support for LPC11C22/C24 added

V2.12

- LPC1111/102 and LPC1111/202 Part IDs corrected

V2.13

- LPC11xx Part IDs updated

V2.14

- Support for LPC11D14 and LPC12D27 added
- LPC11U1x details updated

V2.15

- LPC1113/301 Part IDs updated

V2.16

- Support for LPC11U12/11U23/11U24 added

V2.17

- LPC1100XL Part IDs added (LPC1111/12/13/14/15 revisions 103/203/303/323/333)

V2.18

- Support for 28 pin LPC1112/14 added
- Support for LPC11E11/11E12/11E13/11E14 added
- Support for LPC11U34/11U35/11U36/11U37 added

V2.19

- Support for LPC11A02/11A04/11A11/11A12/11A13/11A14 added

V2.20

- Support for LPC1111FDH20/002, LPC1112FD20/102, LPC1112FDH20/102, and LPC1112FHN24 added

V2.21

- Support for LPC1101LV/1102LV/1112LV/1114LV added

V2.22

- Support for LPC1833/1837/1853/1857 added

V2.23

- LPC11U3x information updated

V2.24

- Support for LPC4333/4337/4353/4357 added

V2.25

Support for LPC4072/4074/4076/4078/4088 added

V2.26

- Sector 29 info in the LPC175x/6x Flash Memory Sectors and Addresses table updated
- Support for LPC1315/1316/1317/1345/1346/1347 added

V2.27

- List of supported 18xx devices updated
- Details on Pins Used in LPC181x/2x/3x/5x and LPC433x/5x Parallel Programming updated

V2.28

- List of supported LPC18xx and LPC43xx devices updated

V2.30

- Support for LPC1765 and LPC1768 TFBGA100 added

PART IDENTIFICATION

LPC1111 is available in a 20 pin TSSOP package. LPC1112 is available in 20 pin SO and TSSOP packages as well as a 24 pin HVQFN package. LPC1112/14 are available in a 28 pin TSSOP package, LPC1114 is available in a 28 pin DIP package, LPC1111/12/13/14 are available in a 33 pin HVQFN package, LPC1114 is available in a 44 pin PLCC and a 48 pin TFBGA package and LPC1113/14/15 are available in a 48 pin LQFP package.

LPC1112/14LV are available in a 24 pin HVQFN package, LPC1101/02LV are available in a 25 pin CSP package, and LPC1112/14LV are available in a 33 pin HVQFN package.

LPC11A02/04 are available in a 20 pin CSP package, LPC11A11/12/13/14 are available in a 33 pin HVQFN package, and LPC11A12/14 are available in a 48 pin LQFP package.

LPC11C12/C14/C22/C24 are available in a 48pin LQFP package.

LPC11D14 is available in a 100 pin LQFP package.

LPC11E11/E14 are available in a 33 pin HVQFN package, LPC11E12/E13/E14 are available in a 48-pin LQFP package and LPC11E14 is available in a 64 pin LQFP package.

LPC11U12/U14/U24/U34/U35 are available in a 33 pin HVQFN package, LPC11U12/U13/U14/U23/U24/U34/U35/U36/U37 are available in a 48-pin LQFP package, LPC11U14/U24/U35 are available in a 48-pin TFBGA package, and LPC11U24/U35/U36/U37 are available in a 64-pin LQFP package.

LPC1224/5/6/7 microcontrollers are available in a LQFP48 and a LQFP64 package.

LPC12D27 is available in a 100 pin LQFP package.

LPC1311/13/42/43 microcontrollers are available in a 33 pin HVQFN package and LPC1313/43 microcontrollers are available in a 48 pin LQFP package.

LPC1315/16/17/45/46/47 microcontrollers are available in a 33 pin HVQFN and a 48 pin LQFP package and LPC1317/47 microcontrollers are available in a 64 pin LQFP package.

LPC1751/52/54/56/58/59 microcontrollers are available in a 80 pin LQFP package.

LPC1763/64/65/66/67/68/69 microcontrollers are available in a 100 pin LQFP package.

LPC1765/68 microcontrollers are available in a 100 pin TFBGA package.

LPC1774/78/88 microcontrollers are availeble in a 144 pin LQFP package.

LPC1776/78/88 microcontrollers are available in a 180 pin TFBGA package.

LPC1774/76/77/78/85/86/87/88/ microcontrollers are available in a 208 LQFP package.

LPC1778/88 microcontrollers are available in a 208 TFBGA package.

LPC1812/13/15/17/22/23/25/27/33/37 are available in a 100 pin TFBGA and 144 pin LQFP package. LPC1853/57 are available in a 208 LQFP package. LPC1833/37/53/57 are available in a 256 pin LBGA package.

LPC2101/2102/2103 and LPC2104/2105/2106 microcontrollers are available in a 48 pin LQFP package.

LPC2109/2114/2119/2124/2129/2194, LPC2131/2132/2134/2136/2138, LPC2141/2142/2144/2146/2148 microcontrollers are available in a 64 pin LQFP package.

LPC2212/2214/2294 microcontrollers are available in a 144 pin LQFP package.

LPC2292 microcontroller is available in a 144 pin LQFP package and 144 ball TFBGA package.

LPC2361/2362/2365/2366/2367 microcontrollers are available in a 100 pin LQFP package.

LPC2364/2368 microcontrollers are available in a 100 pin LQFP package and 100 ball TFBGA package.

LPC2377/2378 microcontrollers are available in a 144 pin LQFP package.

LPC2387 microcontroller is available in a 100 pin LQFP package.

LPC2388 microcontroller is available in a 144 pin LQFP package.

LPC2458 microcontroller is available in a 180 ball TFBGA package.

LPC2468/78 microcontrollers are available in a 208 pin LQFP package and 208 ball TFBGA package.

LPC4072/74/78 are available in a 80 pin LQFP package. LPC4078 is available in a 100 pin LQFP package. LPC4074/76/78/88 are available in a 144 pin LQFP package. LPC4076/78/88 are available in a 180 pin TFBGA package. LPC4078/88 are available in a 208 LQFP and a 208 pin TFBGA package.

LPC4312/13/15/17/22/23/25/27/33/37 are available in a 100 pin TFBGA and 144 pin LQFP package. LPC4353/57 are available in a 208 LQFP package. LPC4333/37/53/57 are available in a 256 pin LBGA package.

Parts can be identified by these markings on the top of the package:

LPC1XYZ
XXXXXX
XXYWWR

or

LPC2XYZ
XXXXXX
XXYWWR

XYZ in the first line enables this part of the record to identify the LPC1000/2000 device.

The latest letter in the third line (field ‘R’) will identify the device revision. Field ‘Y’ in this line states the year the device was manufactured (e.g. Y=3 identifies year 2003). Field ‘WW’ states the week the device was manufactured during that year.

LPC11XX/XXL/XXLV/AXX/CXX/DXX/EXX/UXX PINOUT

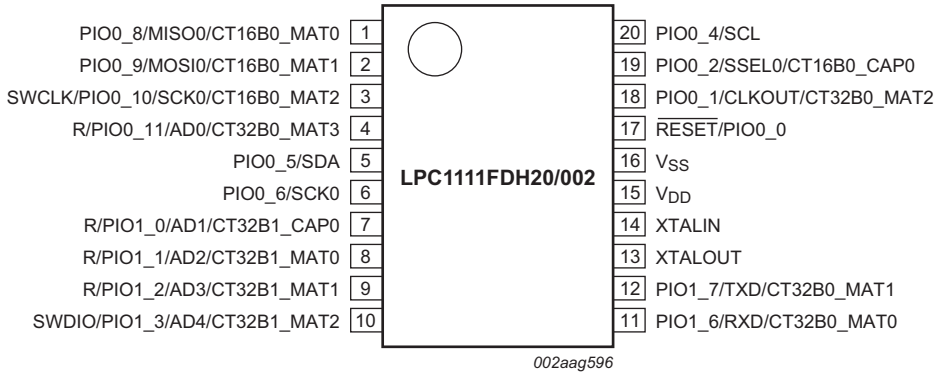


Figure 1: LPC1111 20-pin TSSOP package

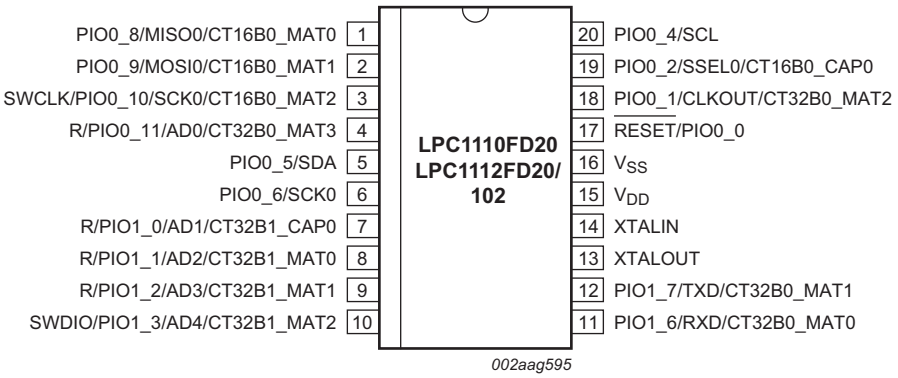


Figure 2: LPC1112 20-pin SO package

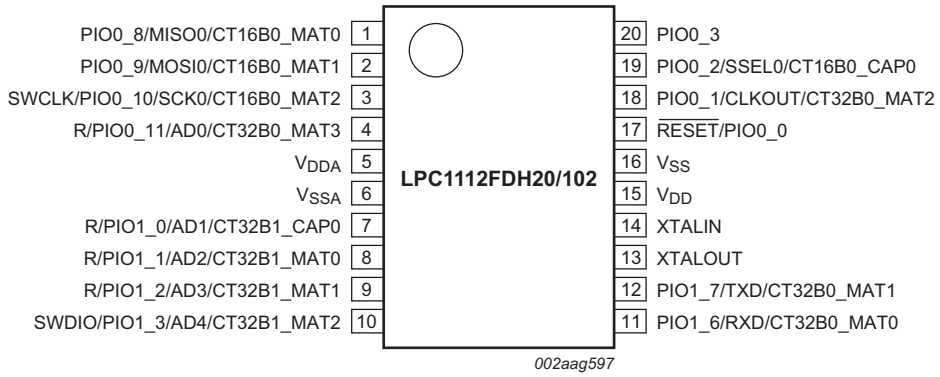


Figure 3: LPC1112 20-pin TSSOP package

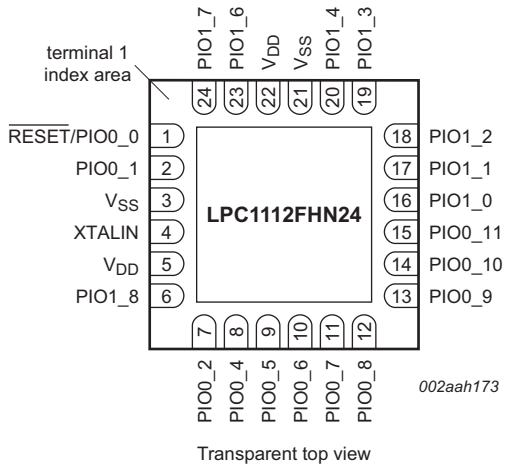


Figure 4: LPC1112 24-pin HVQFN package

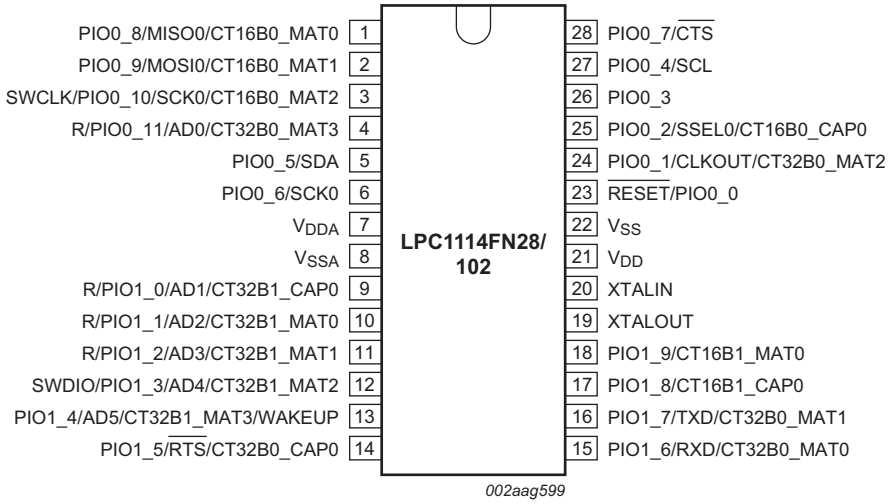


Figure 5: LPC1114 28-pin DIP package

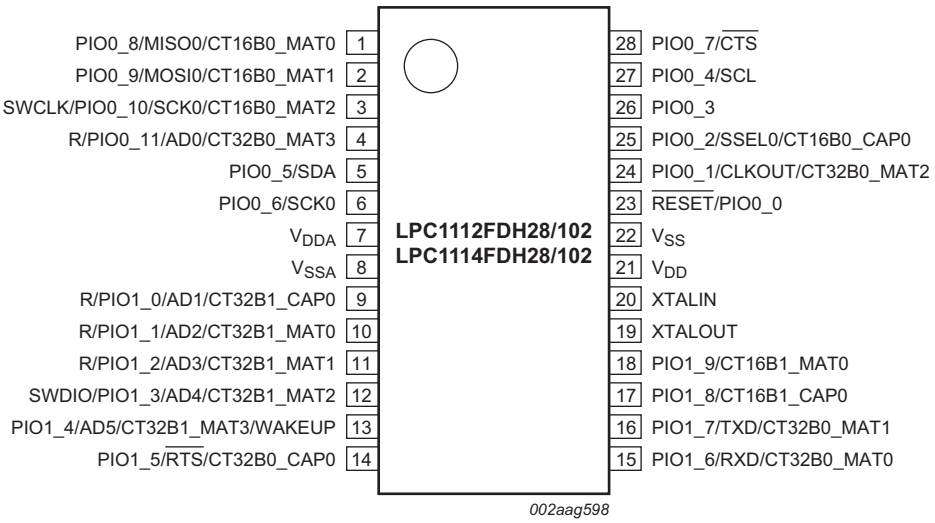


Figure 6: LPC1112/1114 28-pin TSSOP package

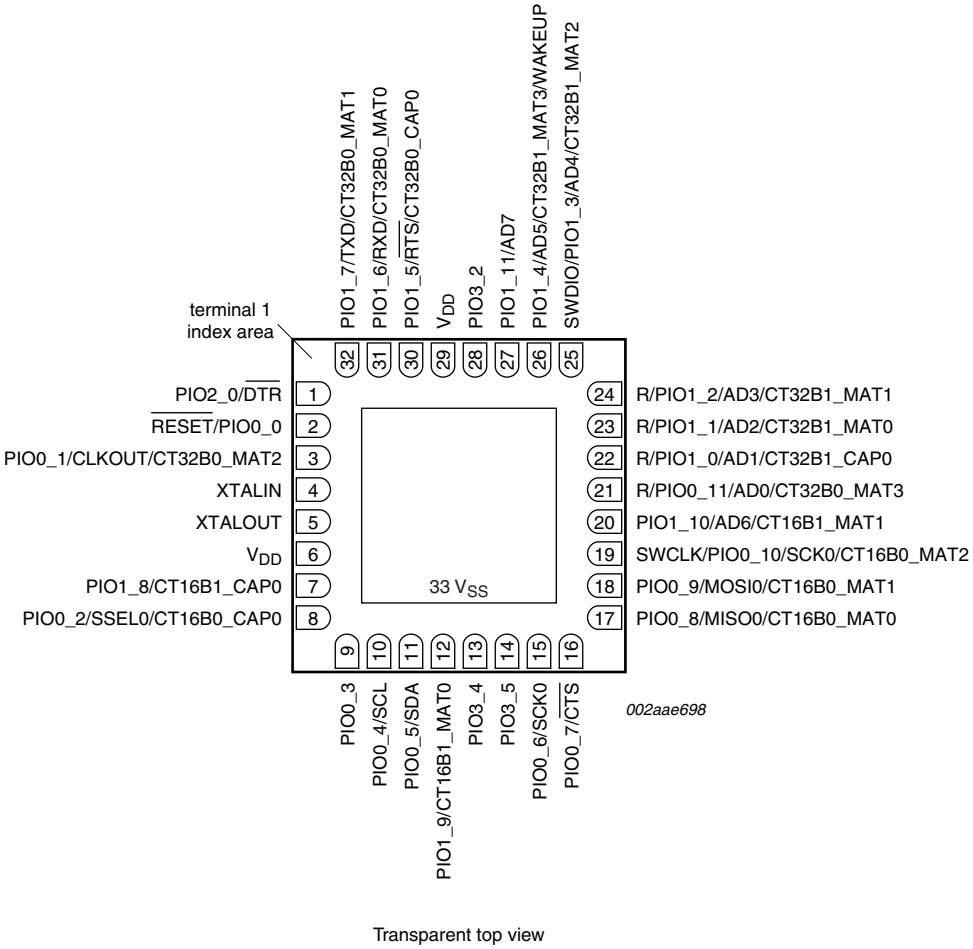


Figure 7: LPC1111/12/13/14 33-pin HVQFN package (LPC1100 and LPC1100L series)

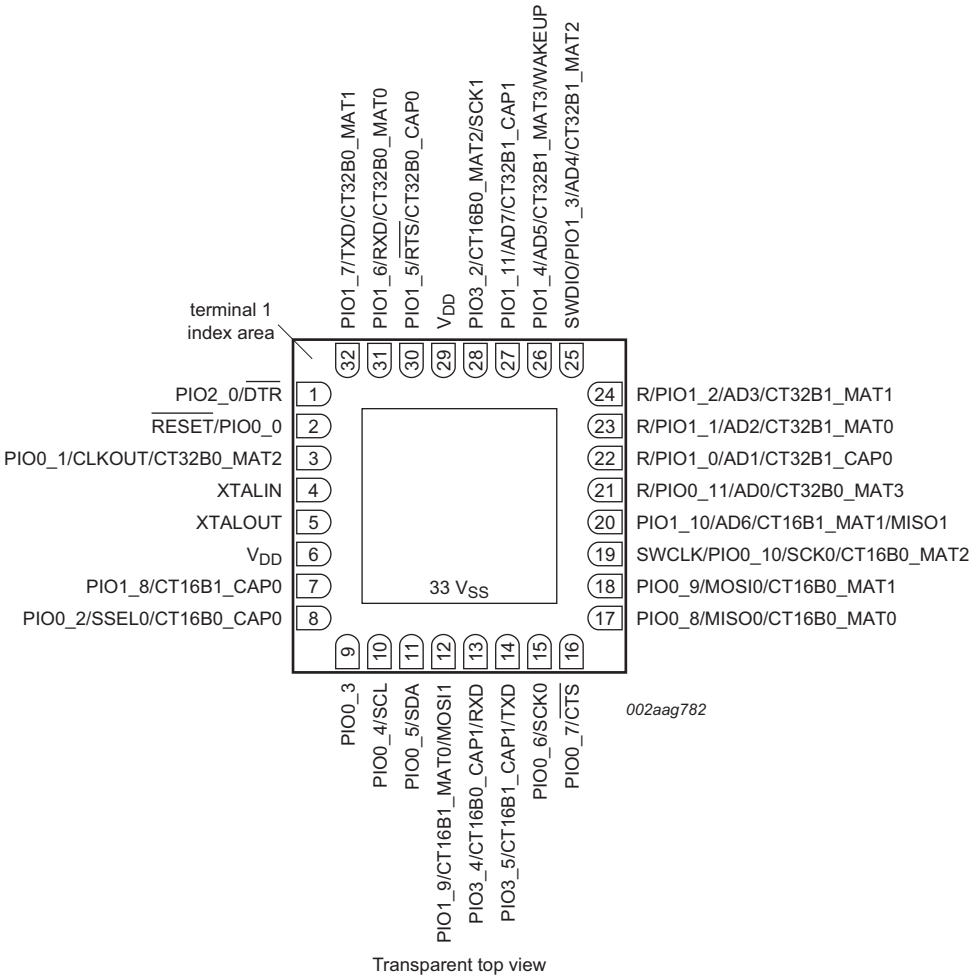


Figure 8: LPC1111/12/13/14 33-pin HVQFN package (LPC1100XL series)

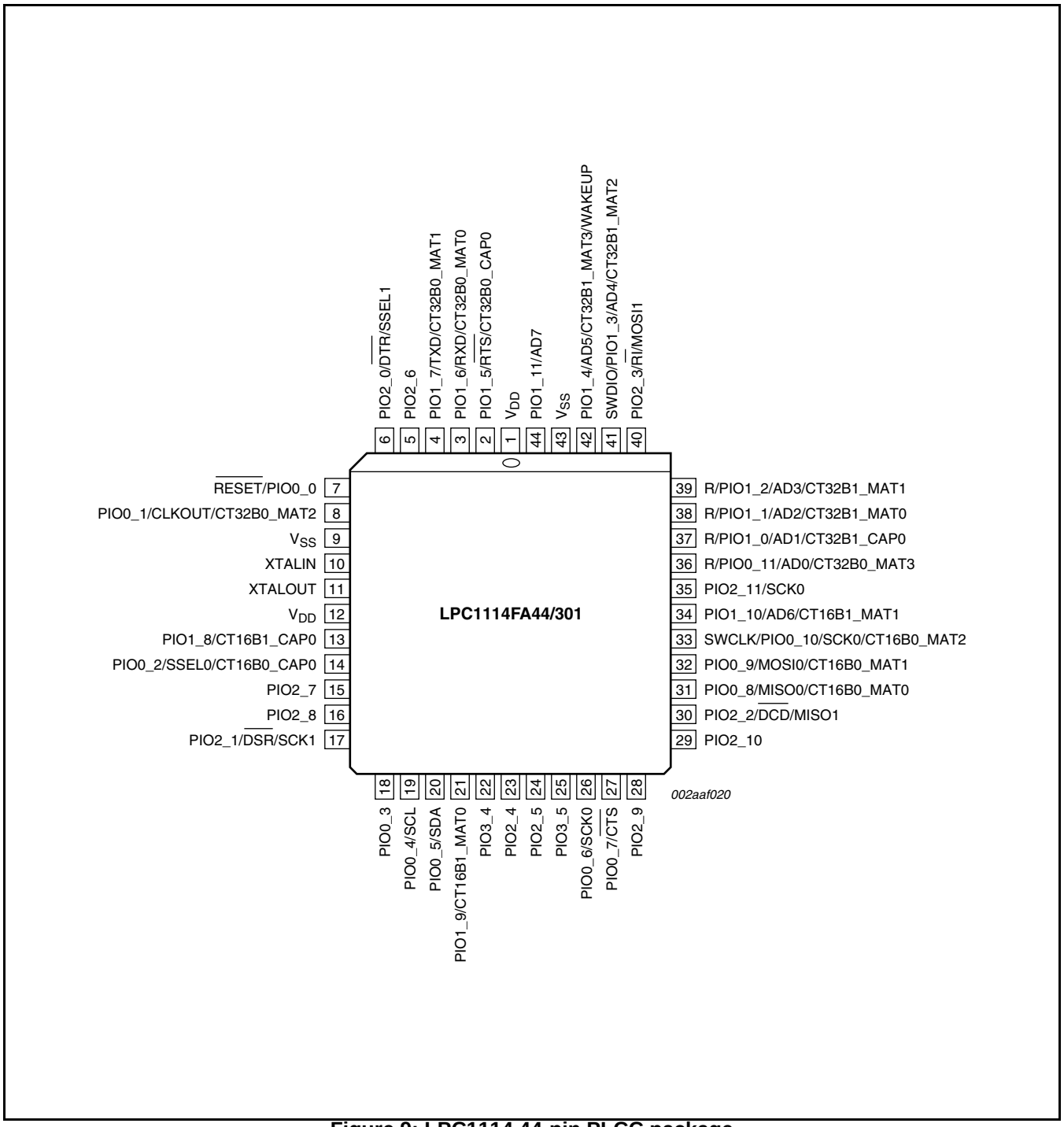


Figure 9: LPC1114 44-pin PLCC package

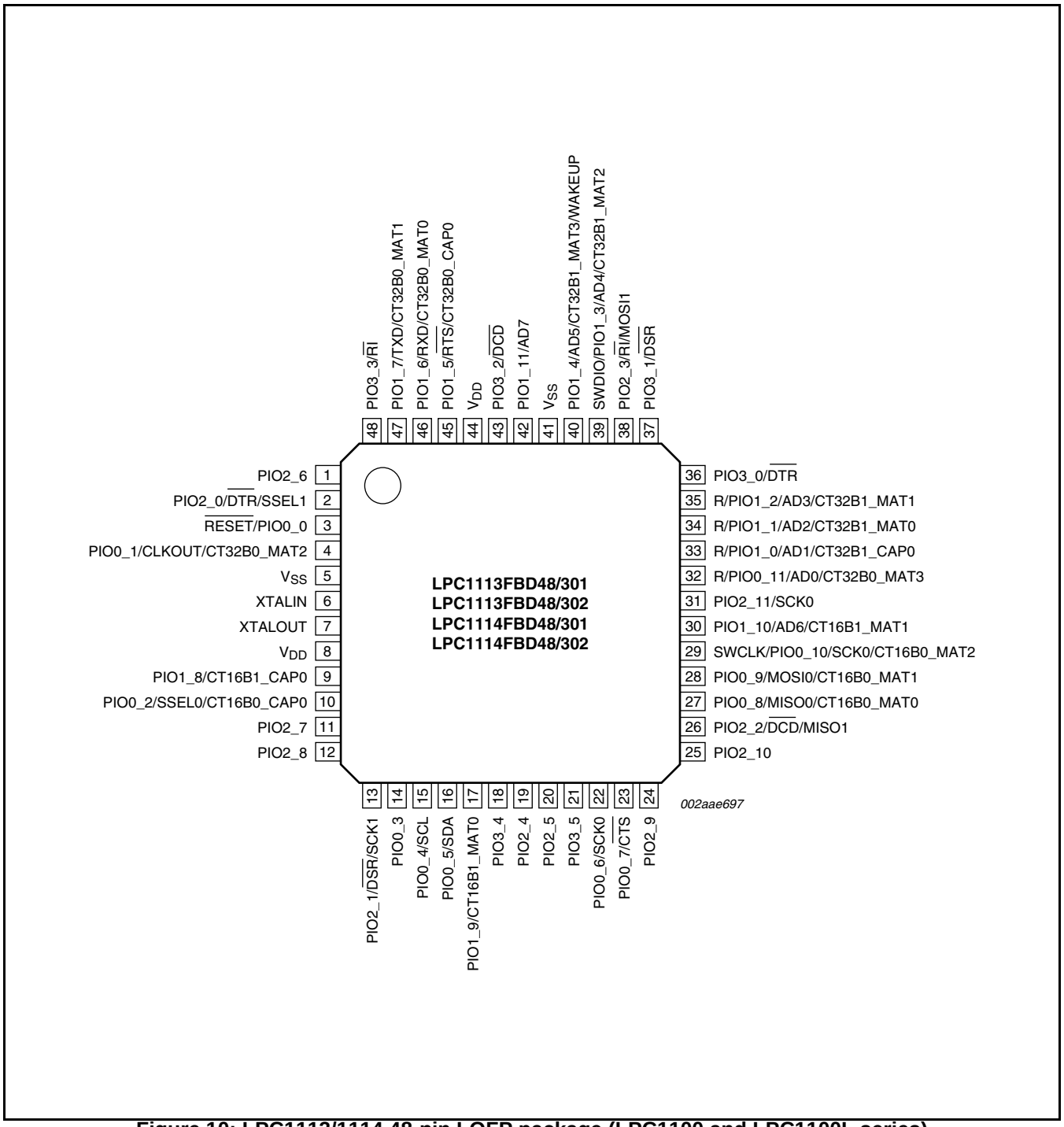


Figure 10: LPC1113/1114 48-pin LQFP package (LPC1100 and LPC1100L series)

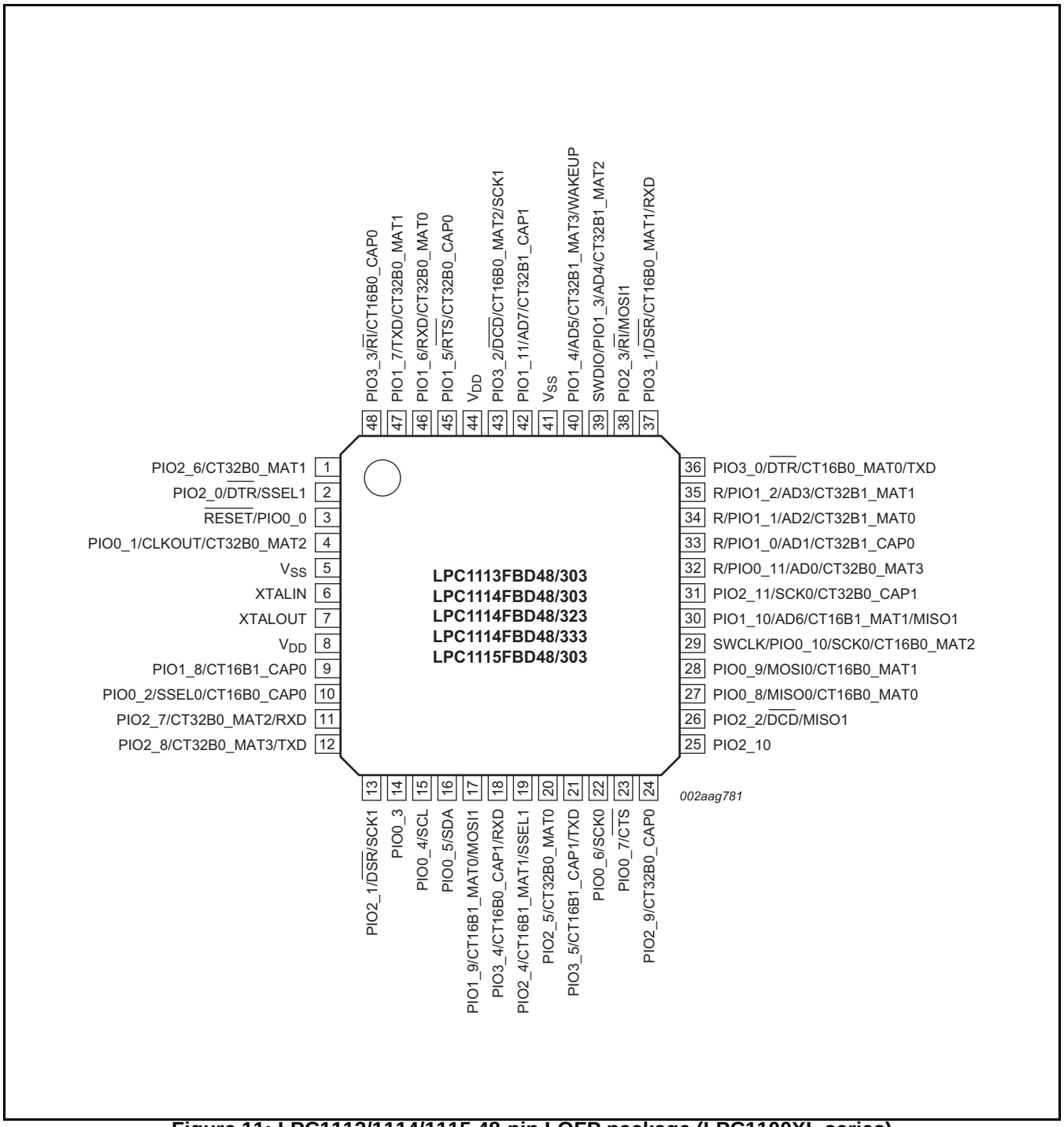


Figure 11: LPC1113/1114/1115 48-pin LQFP package (LPC1100XL series)

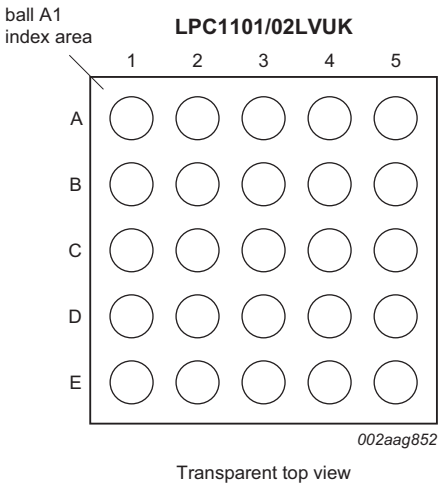


Figure 12: LPC1101LV/1102LV 25-pin WLCSP package

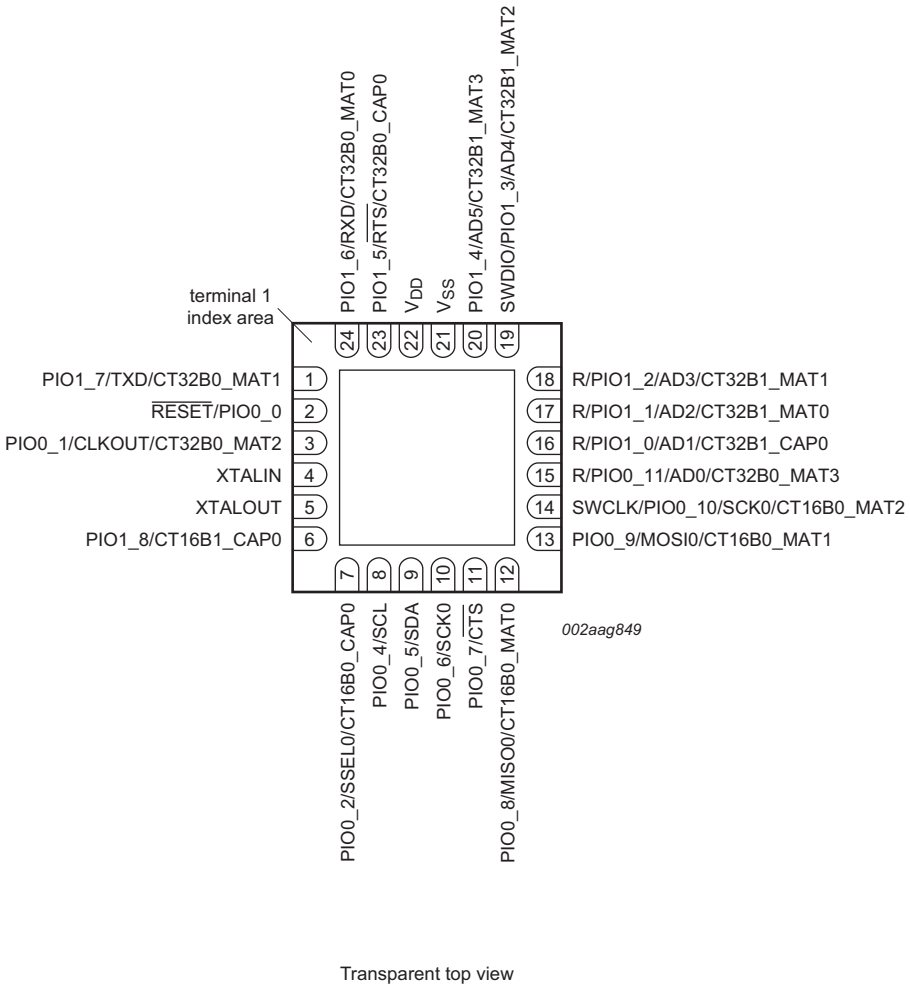


Figure 13: LPC1112LV/1114LV 24-pin HVQFN package

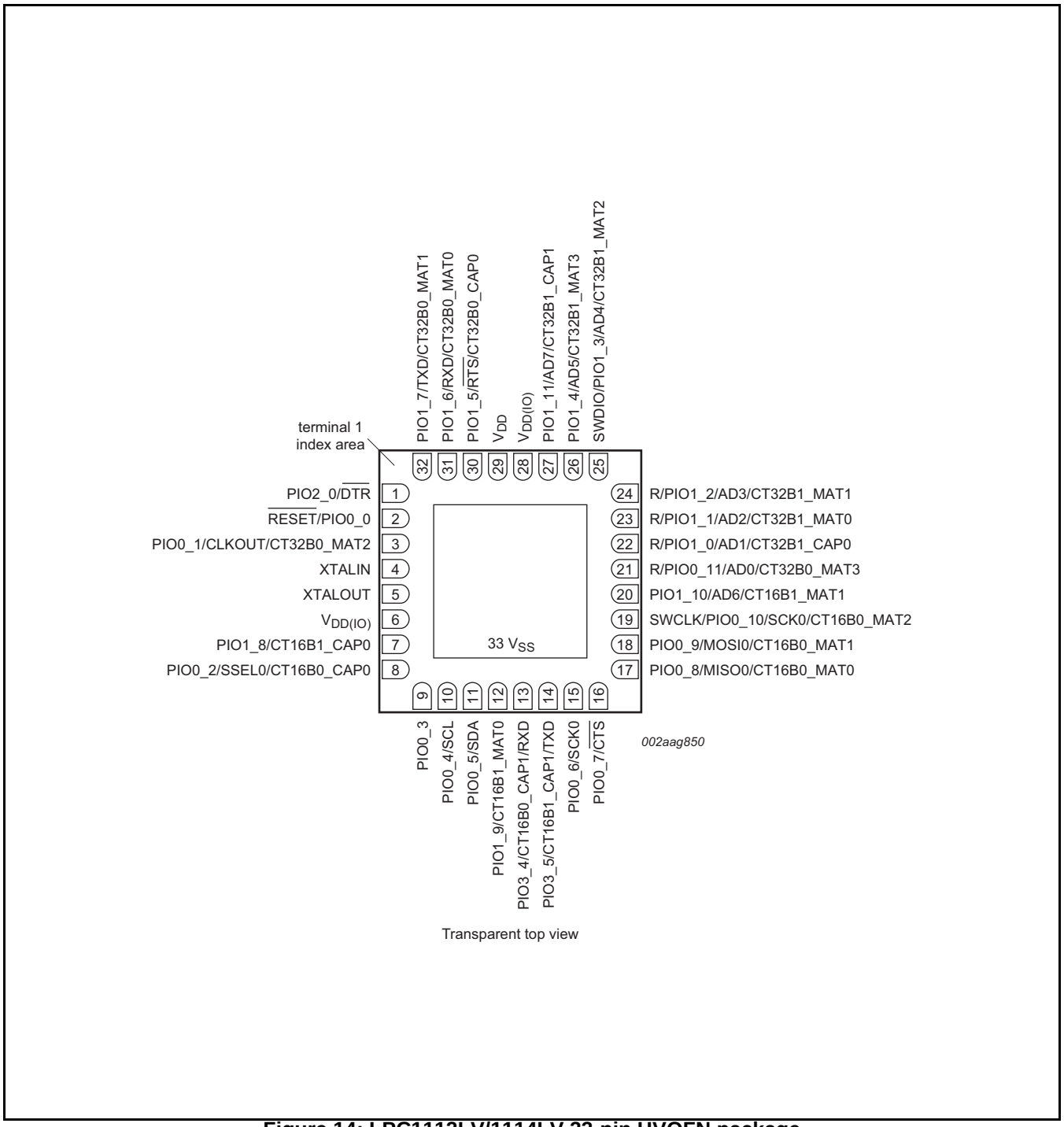


Figure 14: LPC1112LV/1114LV 33-pin HVQFN package

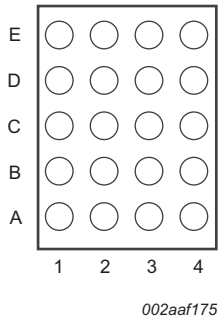


Figure 15: LPC11A02/11A04 20-pin CSP package

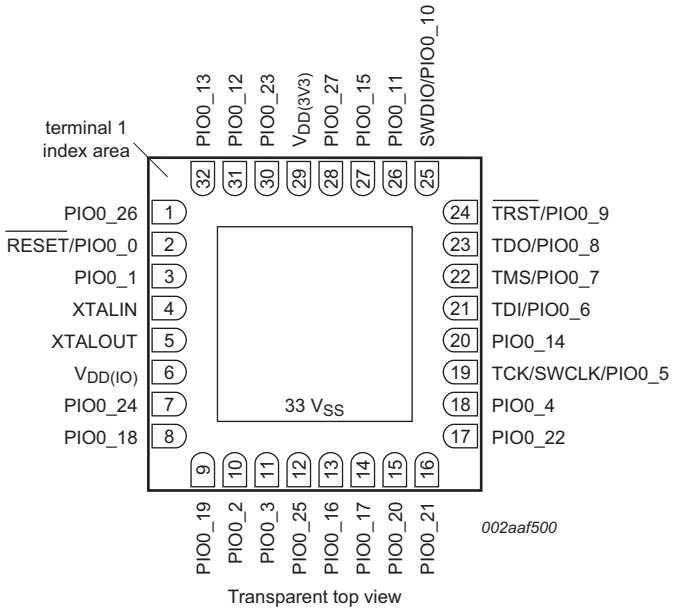


Figure 16: LPC11A11/11A12/11A13/11A14 33-pin HVQFN package

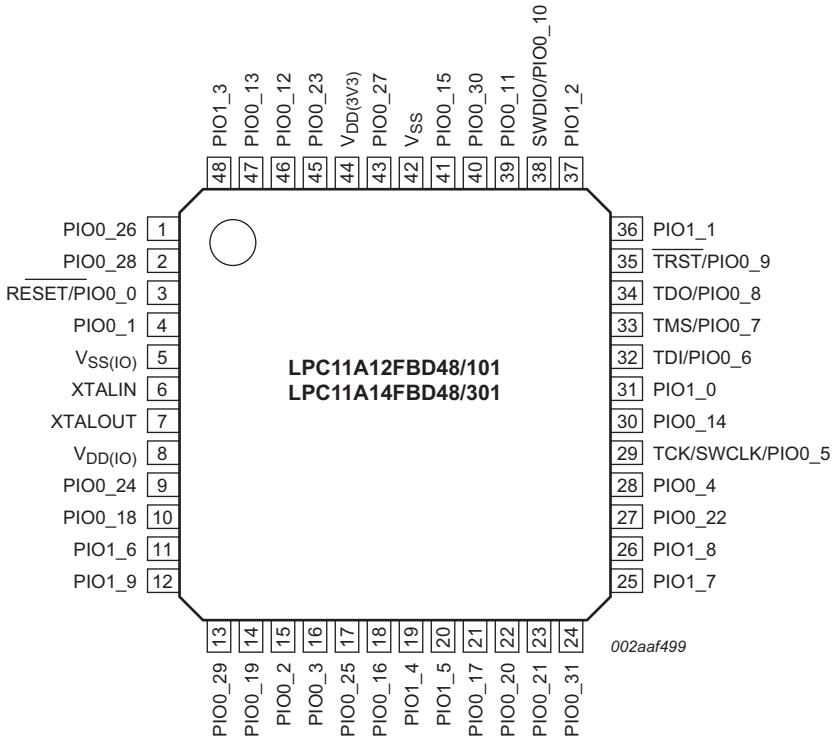


Figure 17: LPC11A11A12/11A14 48-pin LQFP package

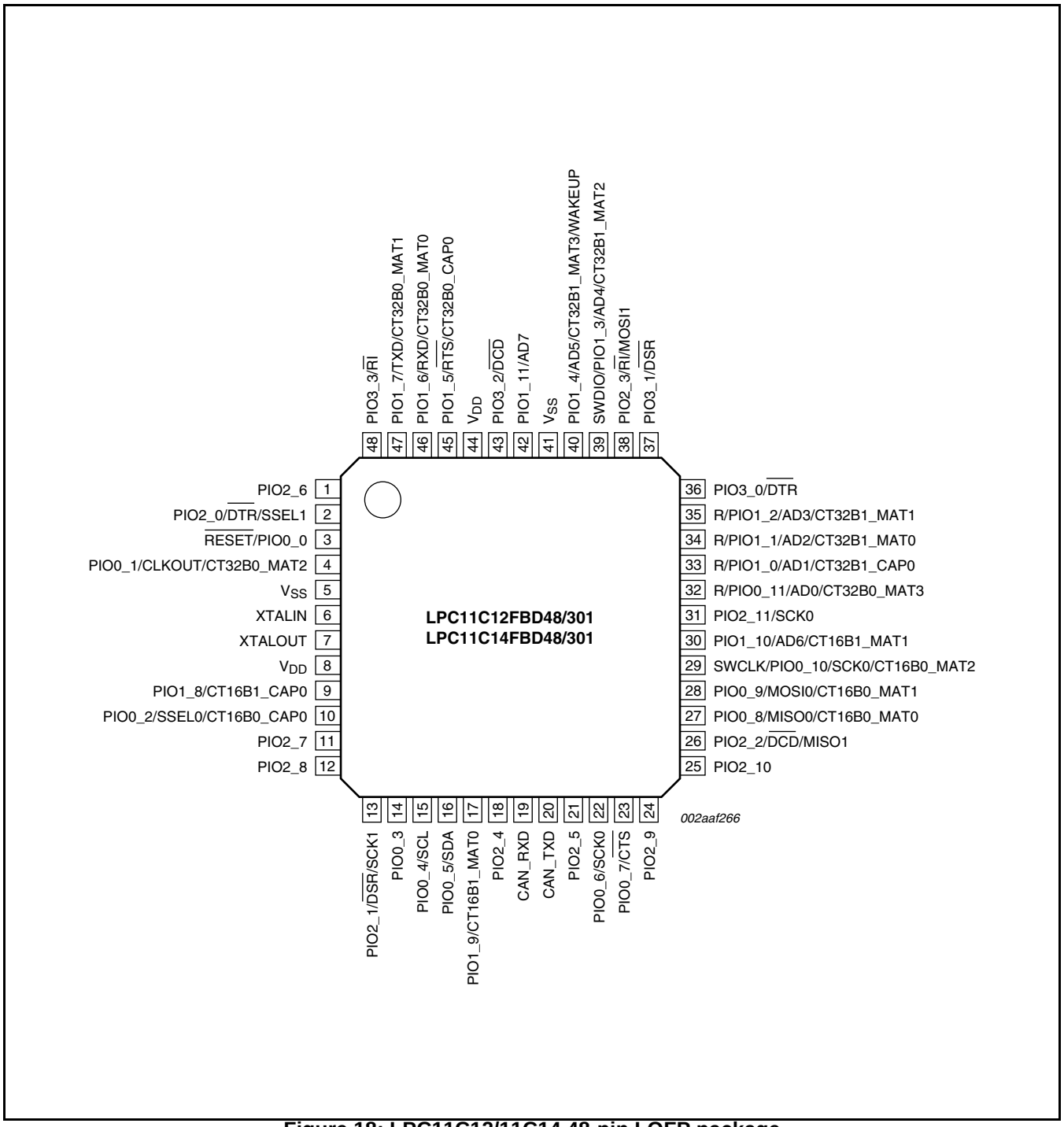


Figure 18: LPC11C12/11C14 48-pin LQFP package

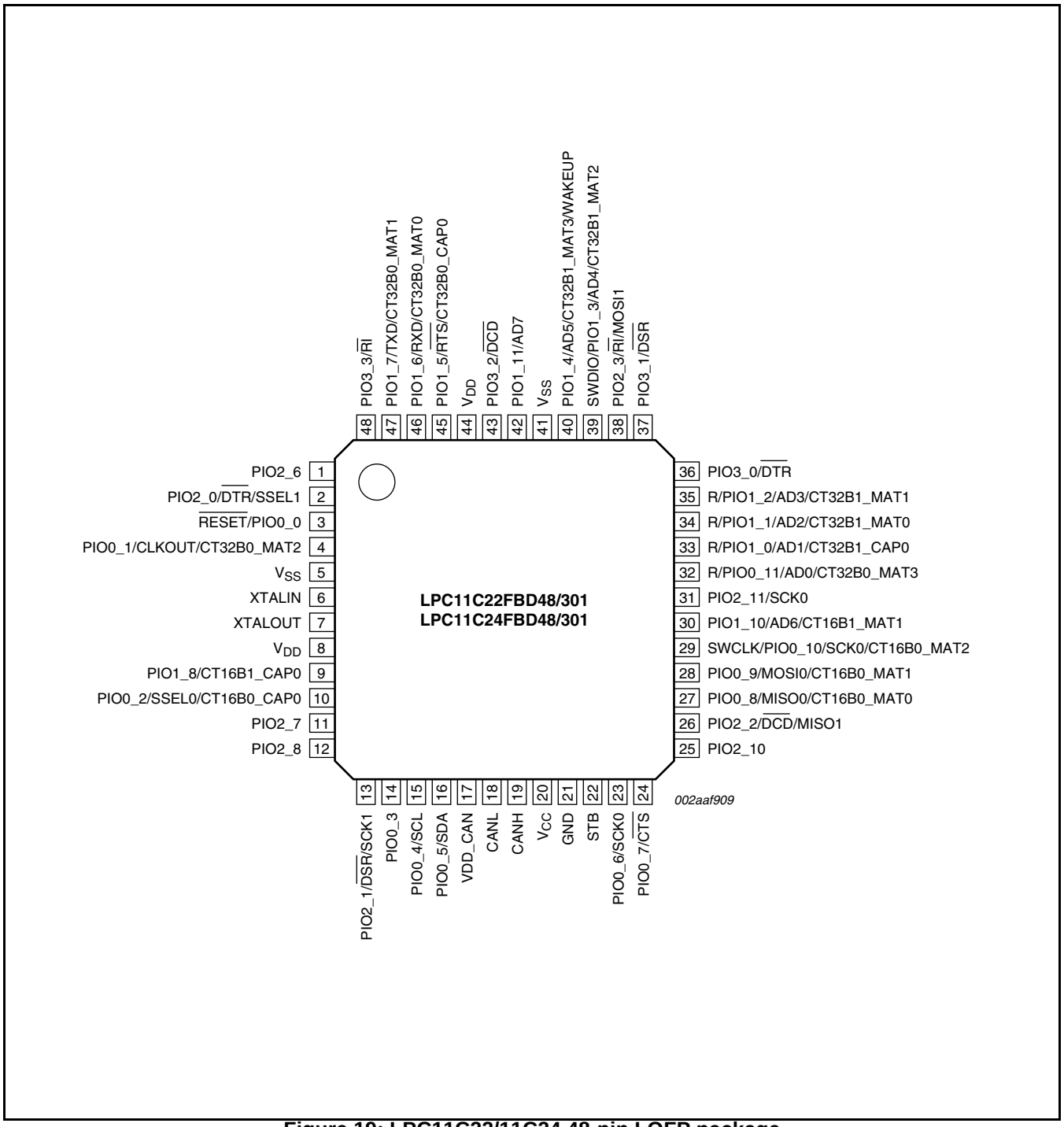


Figure 19: LPC11C22/11C24 48-pin LQFP package

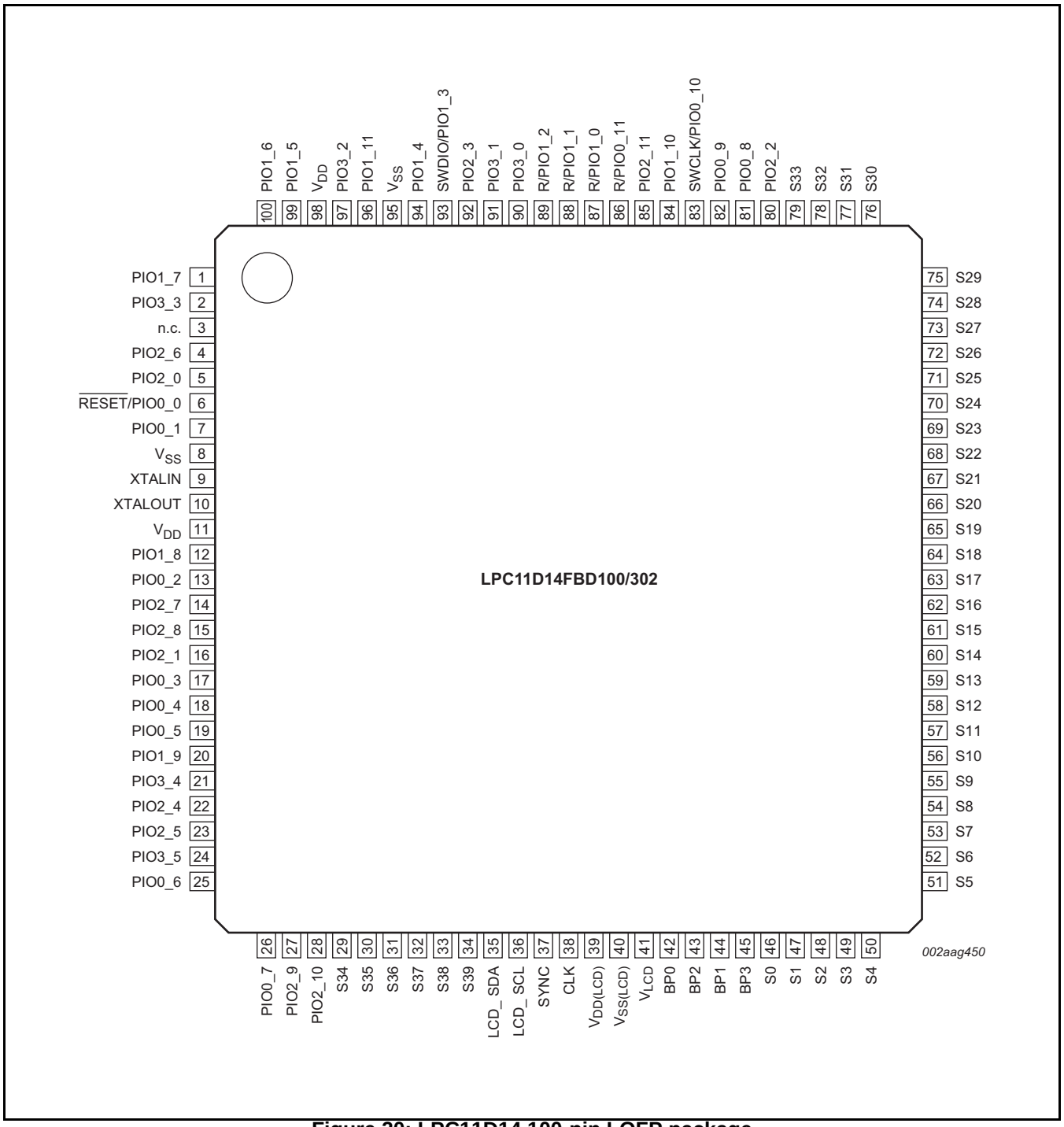


Figure 20: LPC11D14 100-pin LQFP package

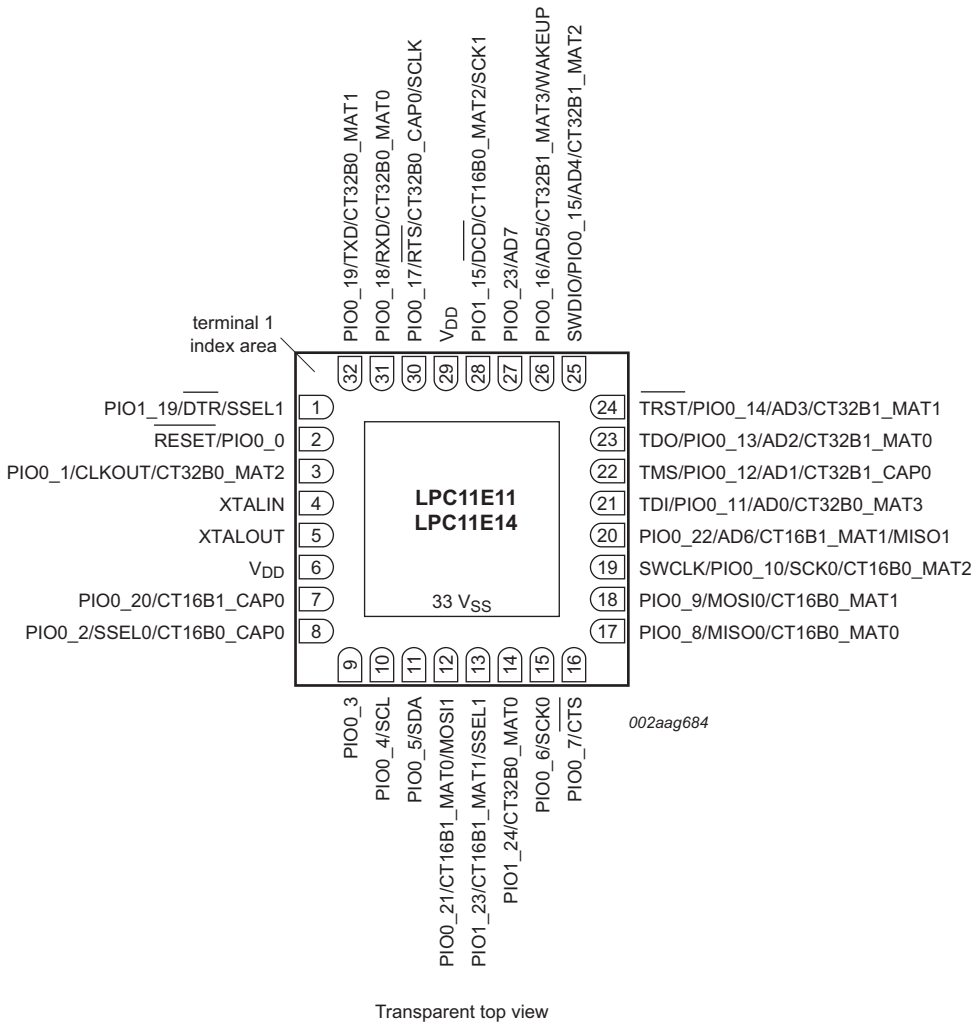


Figure 21: LPCE11/11E14 33-pin HVQFN package

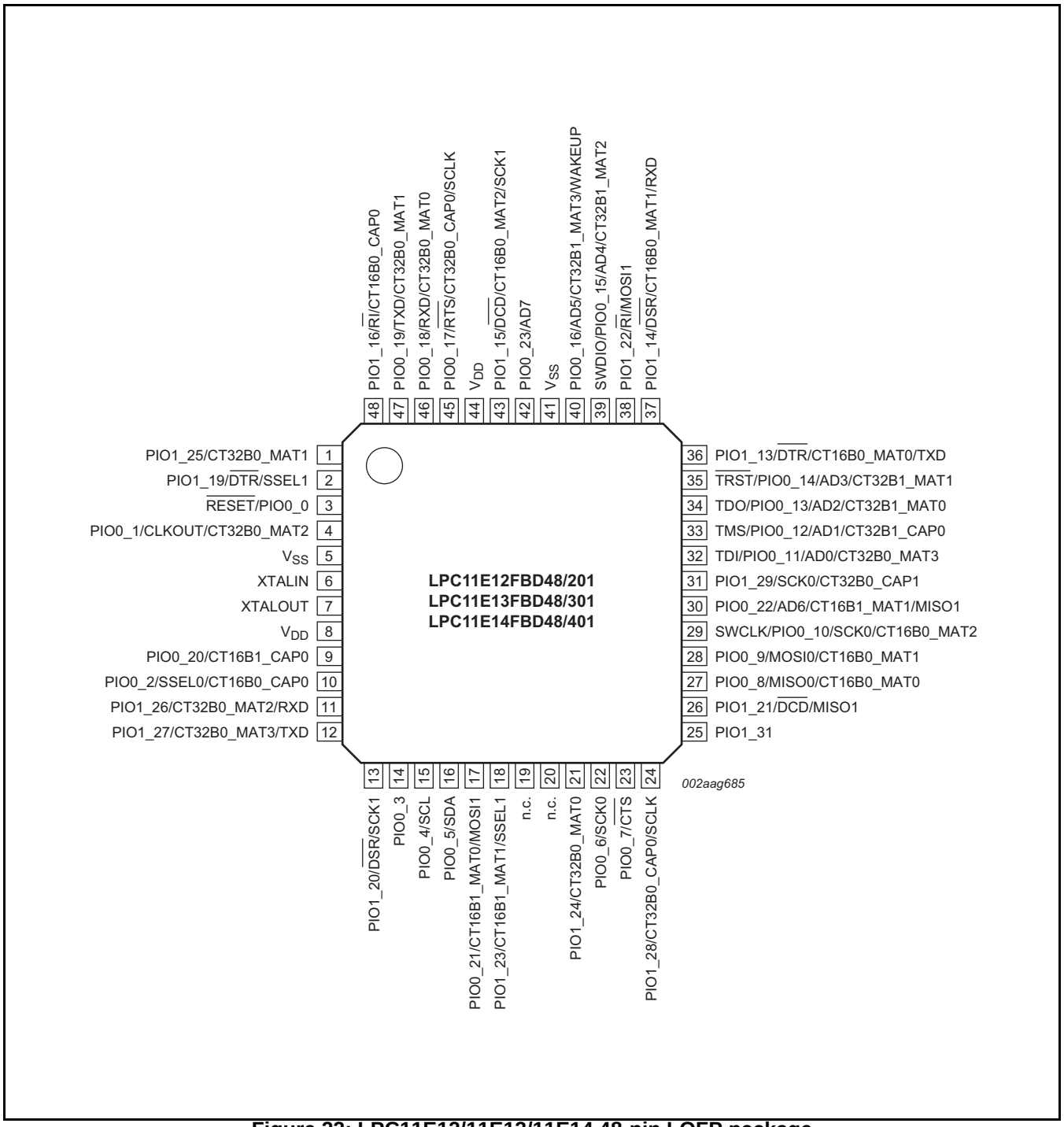


Figure 22: LPC11E12/11E13/11E14 48-pin LQFP package

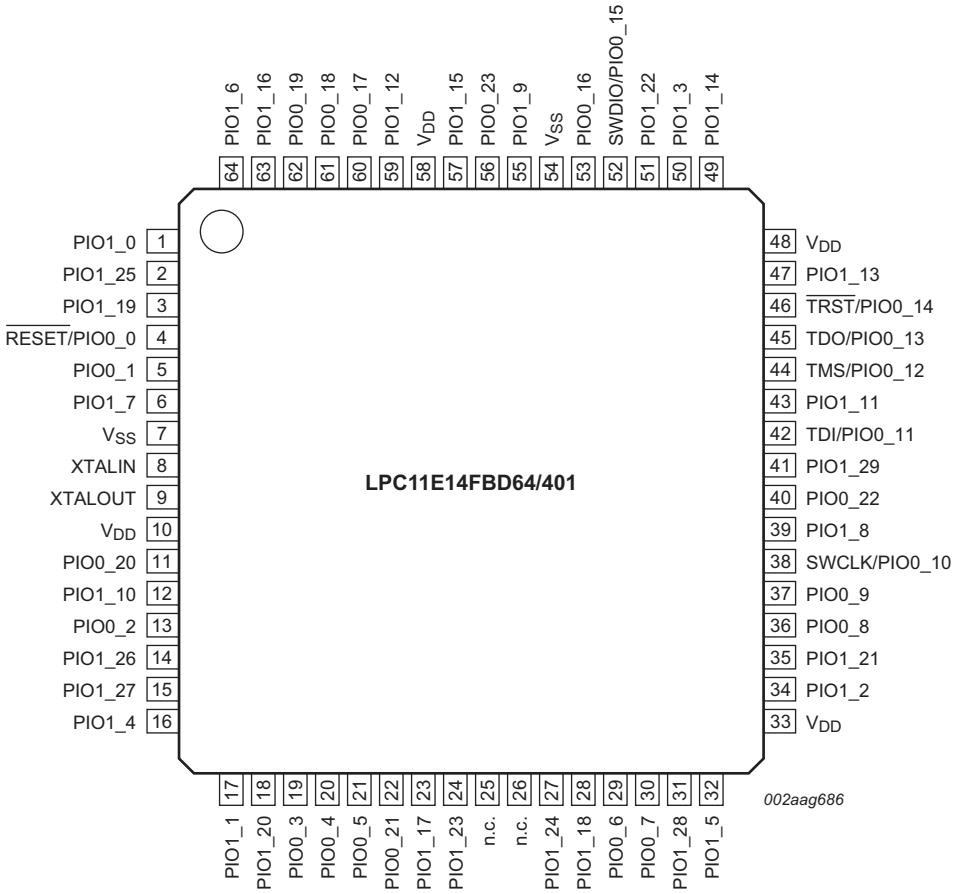


Figure 23: LPC11E14 64-pin LQFP package

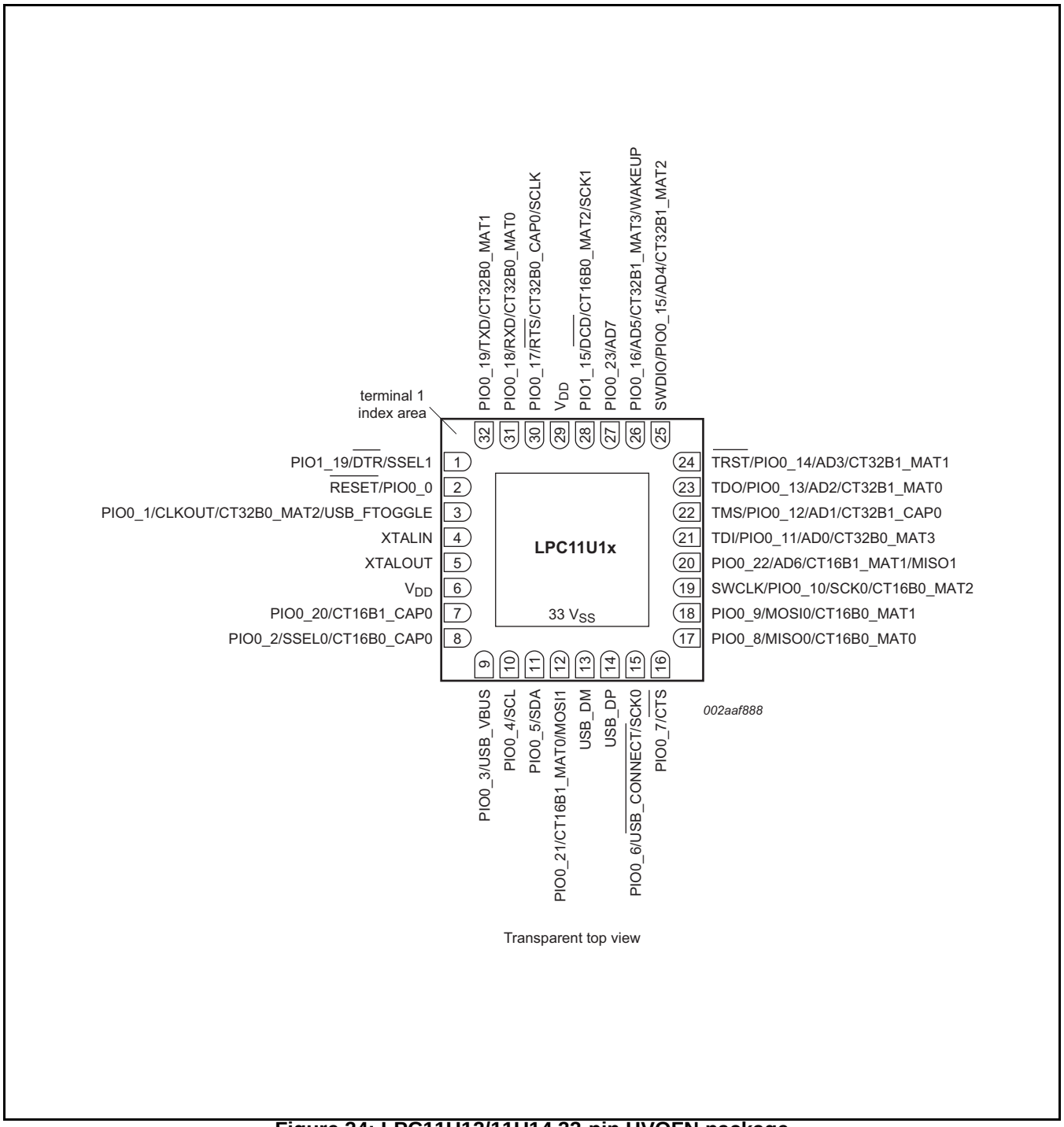


Figure 24: LPC11U12/11U14 33-pin HVQFN package

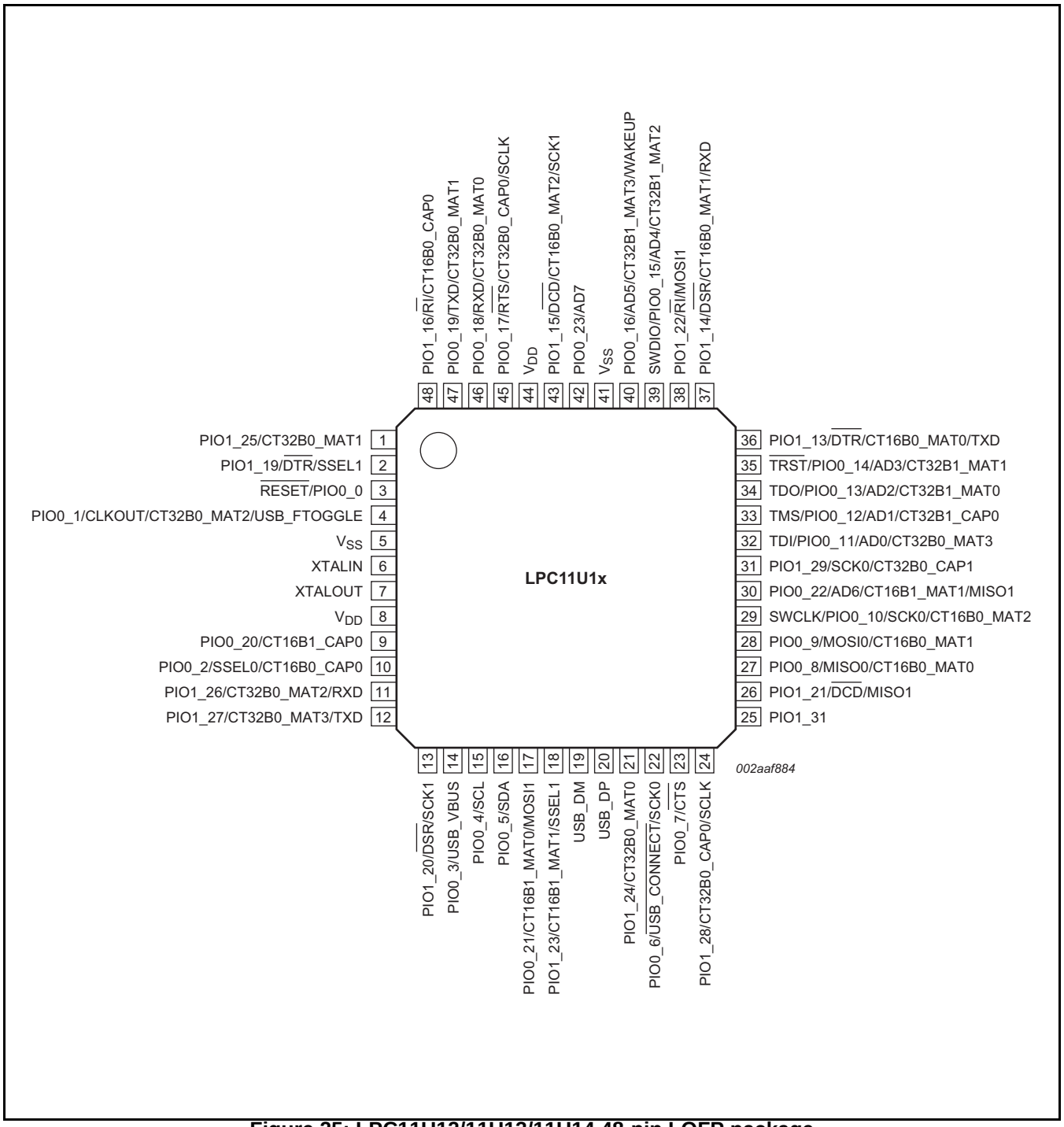


Figure 25: LPC11U12/11U13/11U14 48-pin LQFP package

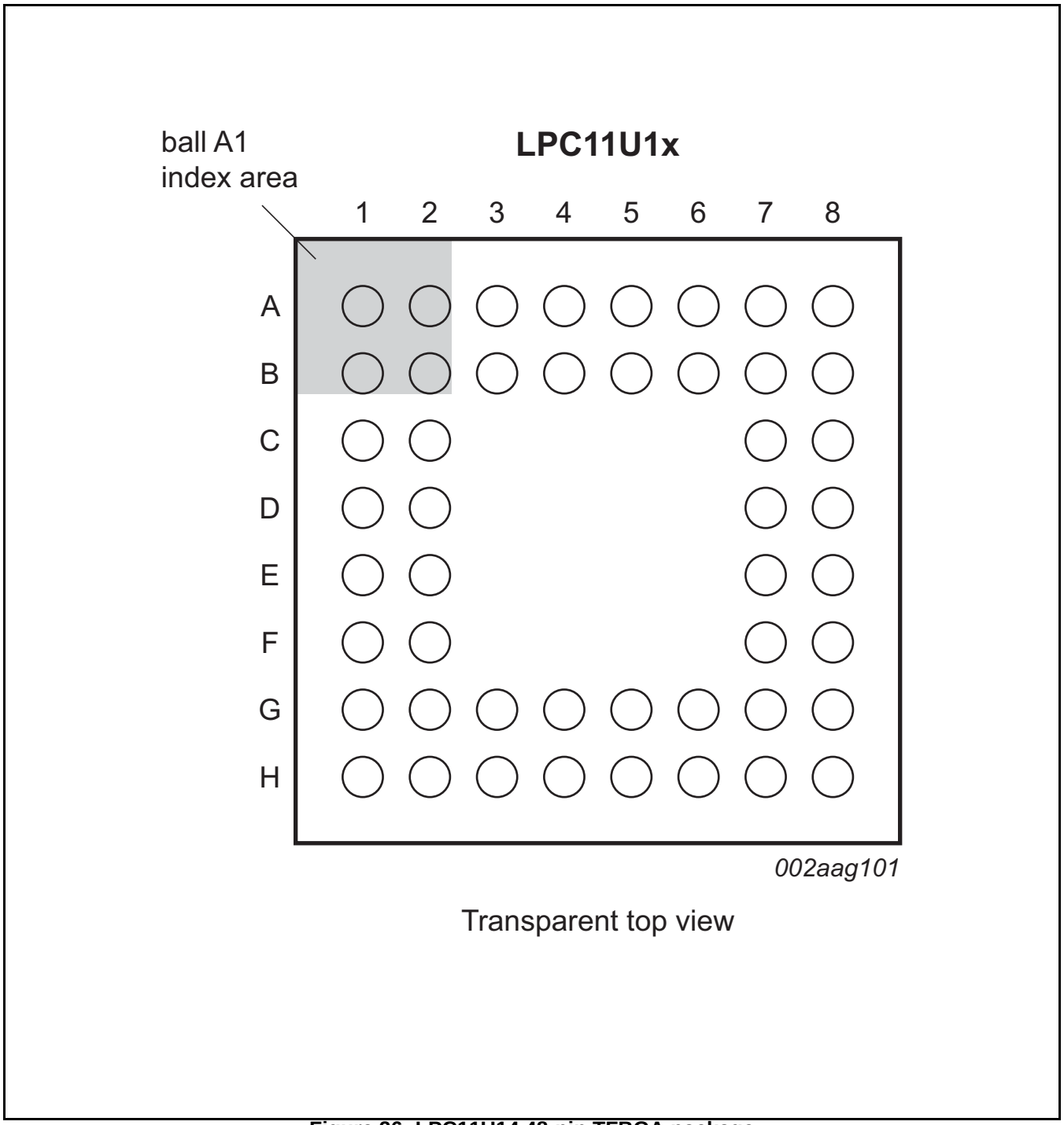


Figure 26: LPC11U14 48-pin TFBGA package

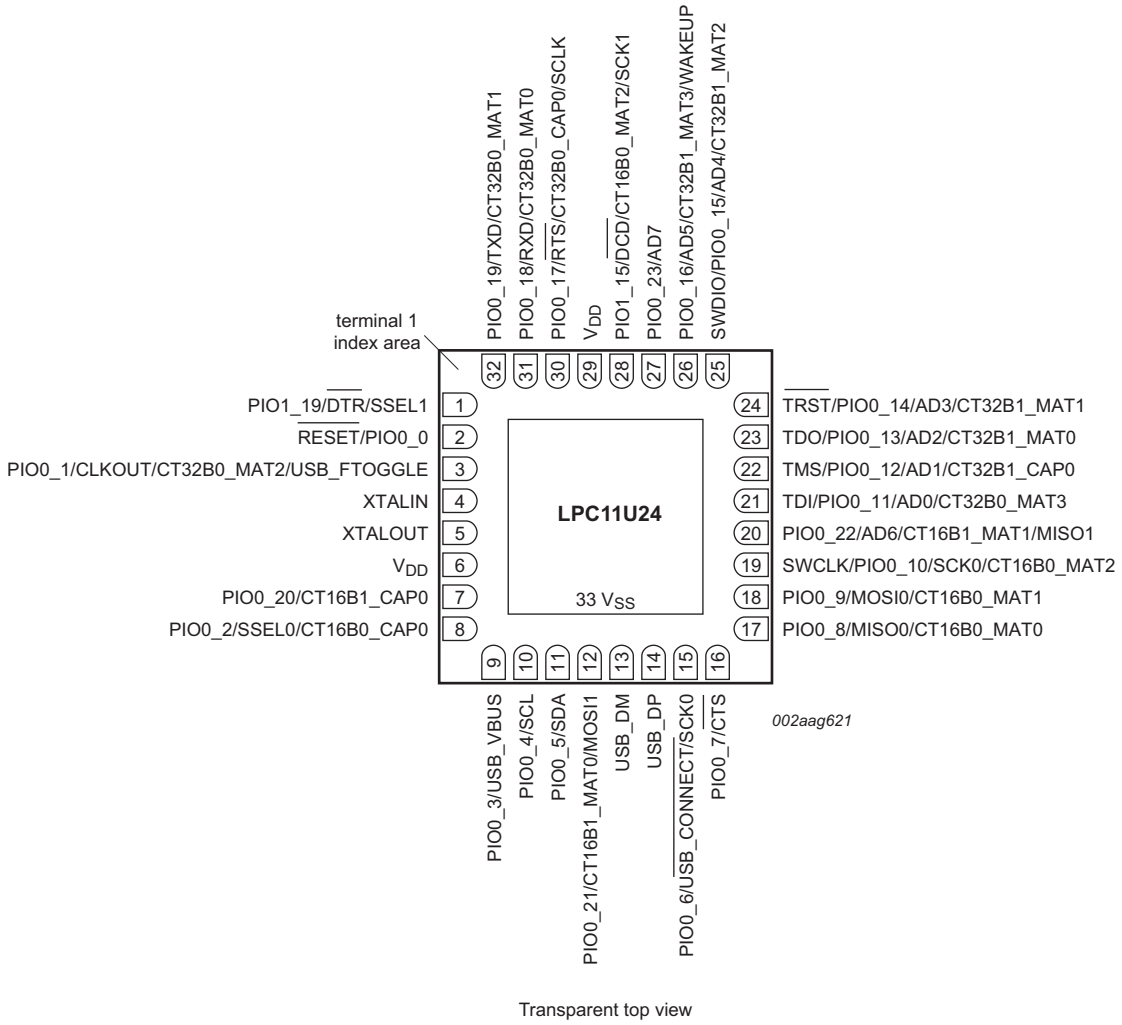


Figure 27: LPC11U24 33-pin HVQFN package

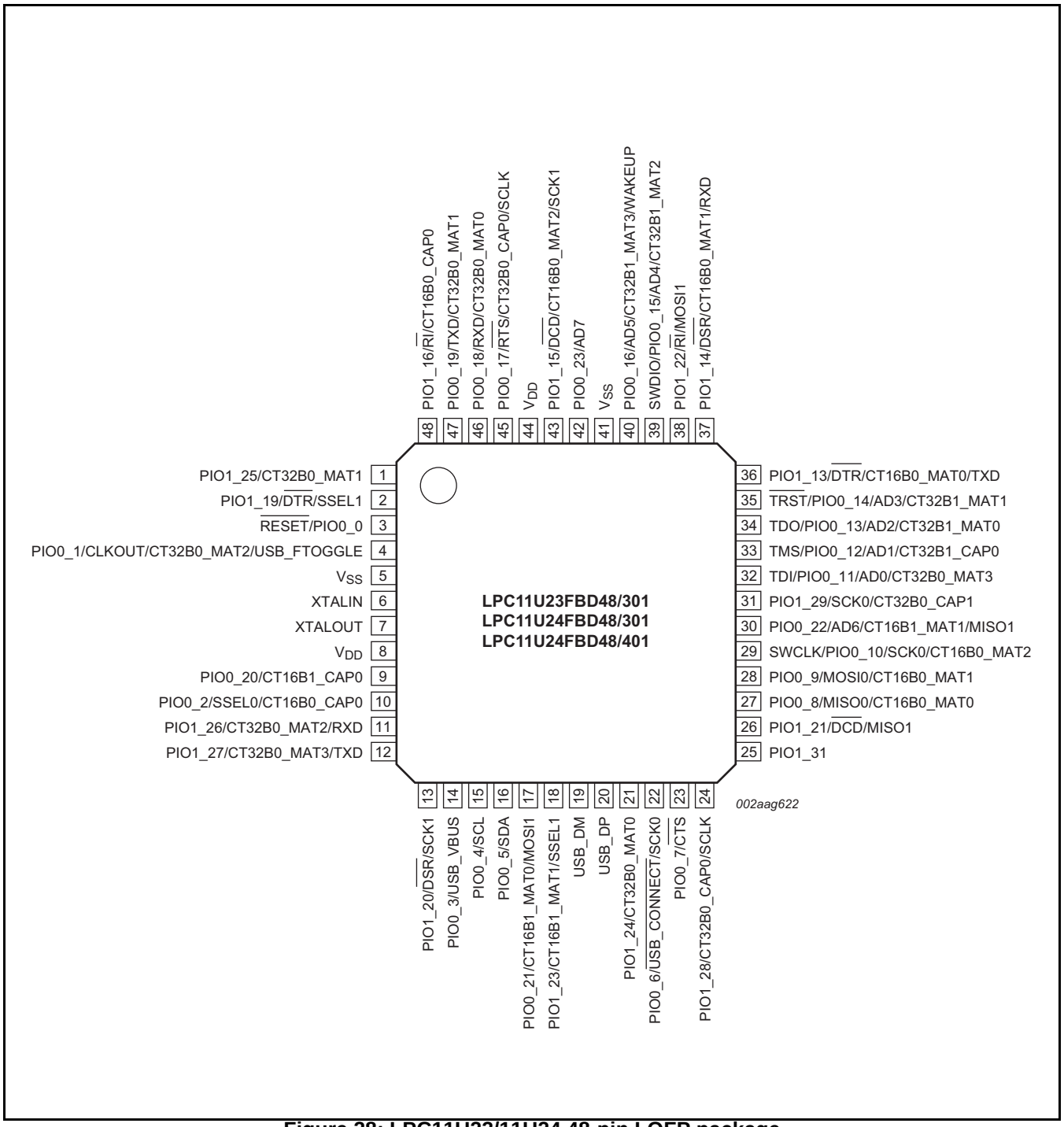


Figure 28: LPC11U23/11U24 48-pin LQFP package

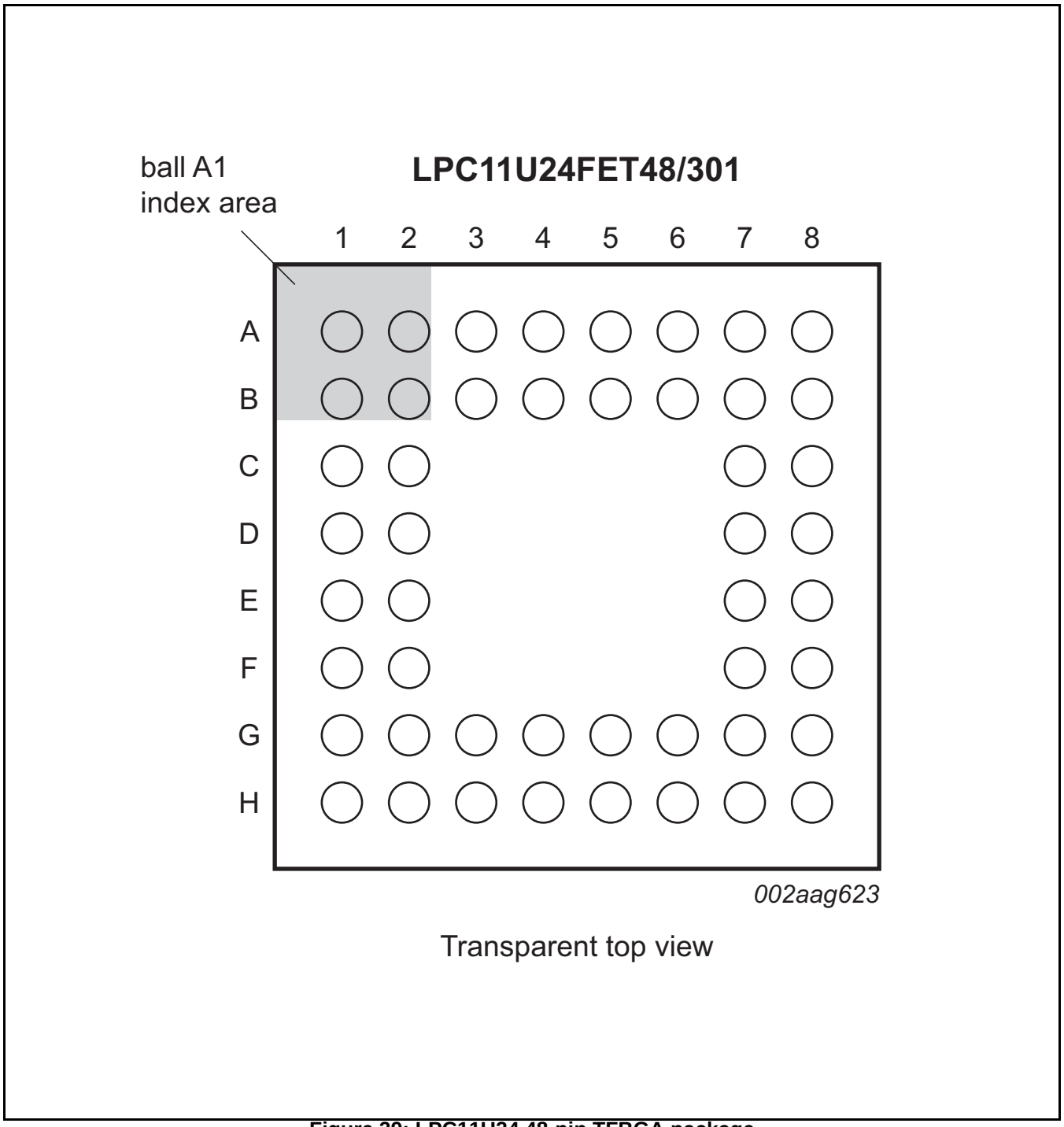


Figure 29: LPC11U24 48-pin TFBGA package

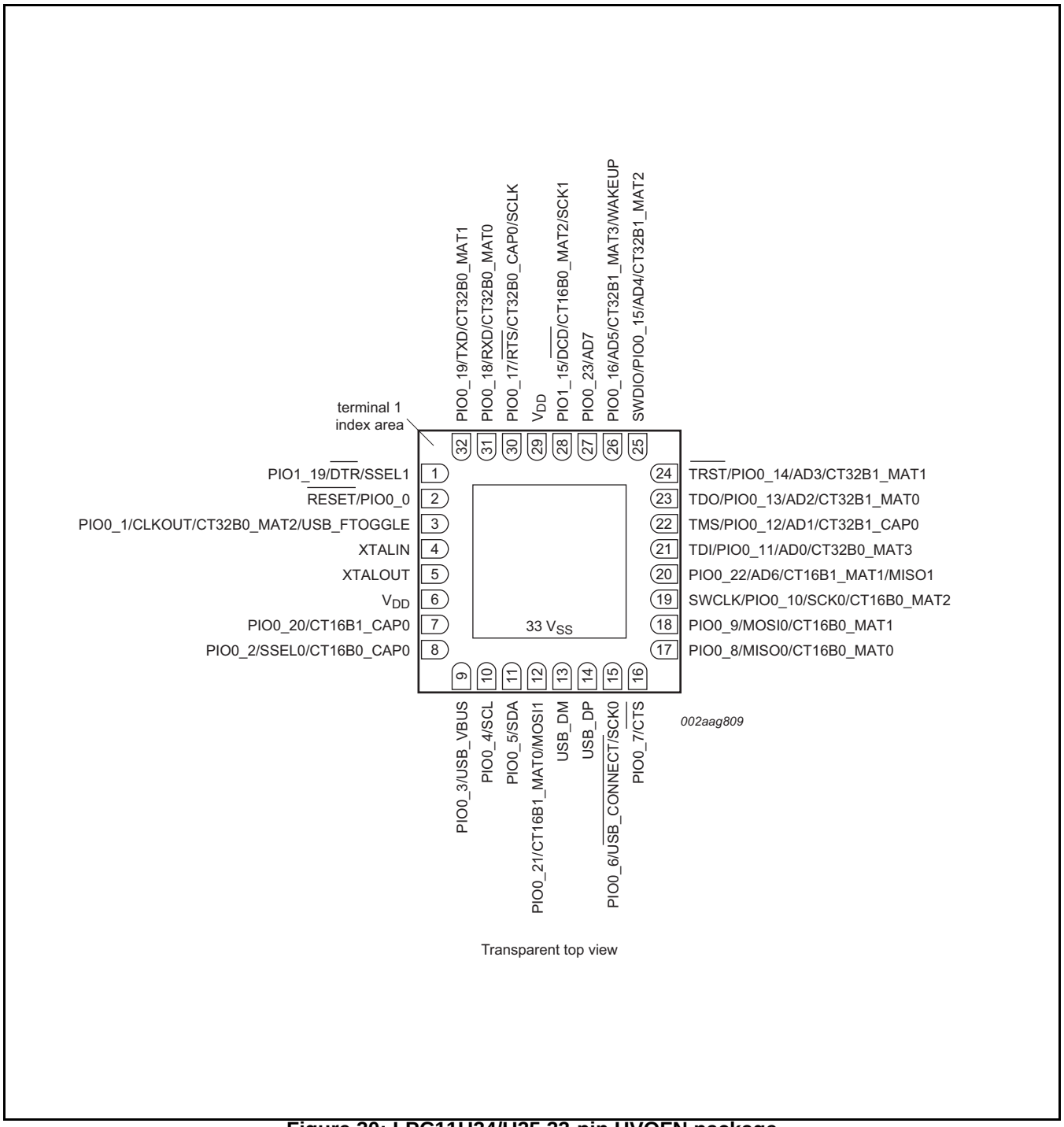


Figure 30: LPC11U34/U35 33-pin HVQFN package

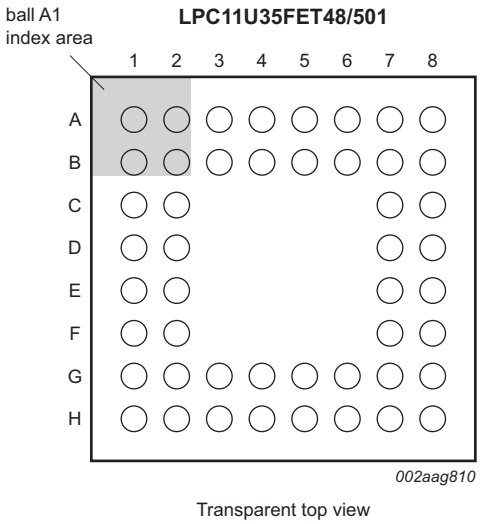


Figure 31: LPC11U35 are available 48-pin TFBGA package

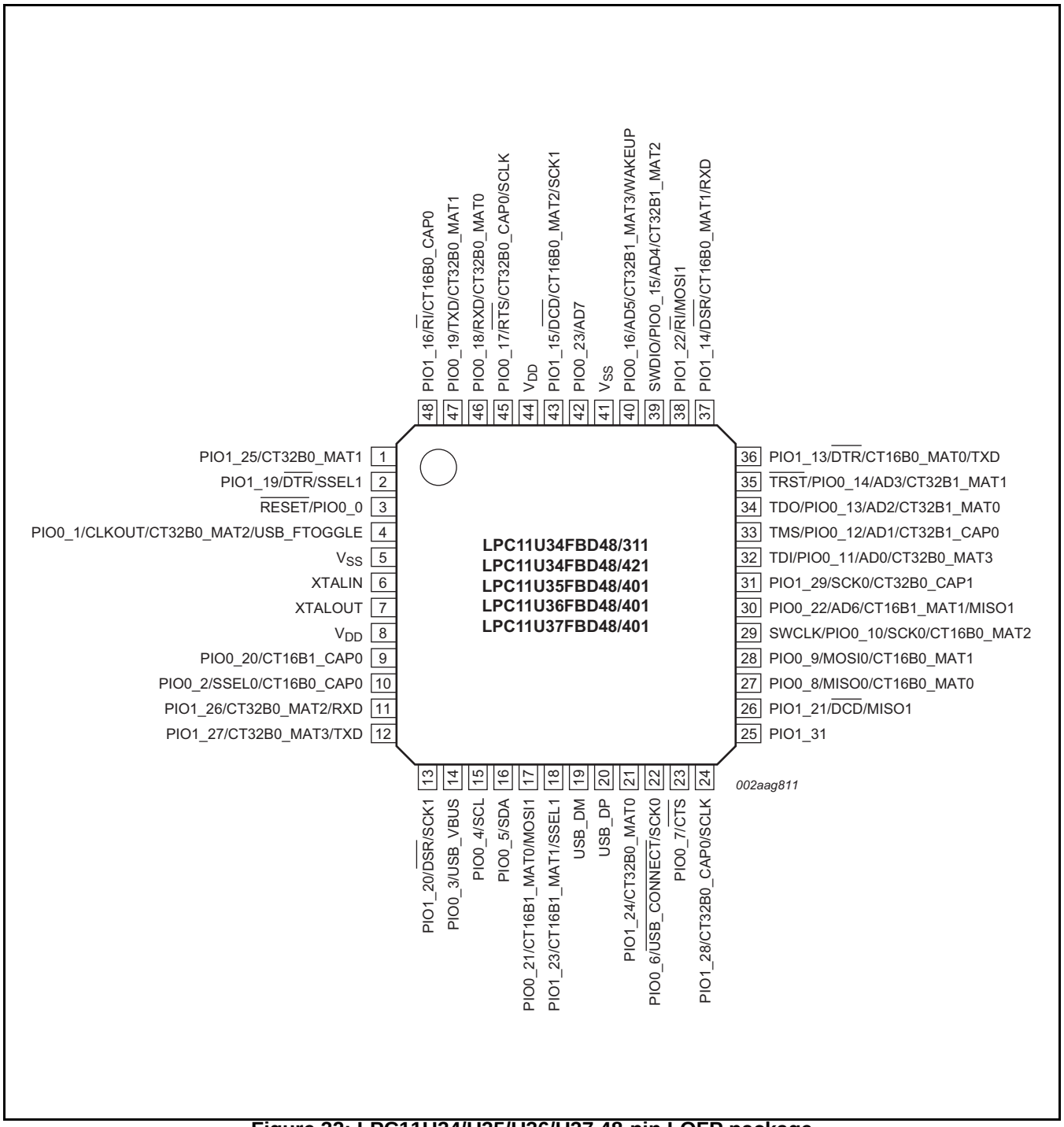


Figure 32: LPC11U34/U35/U36/U37 48-pin LQFP package

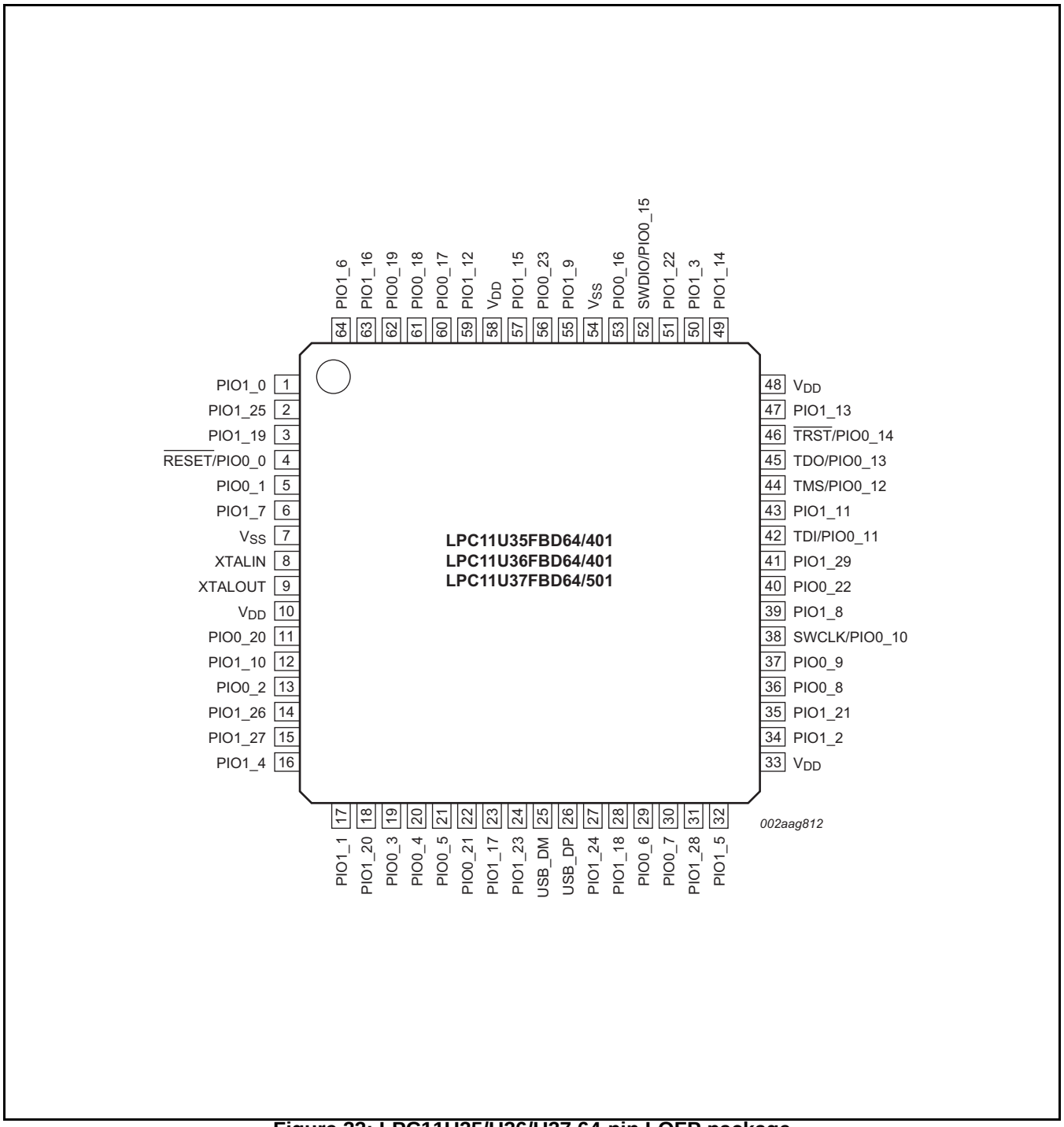


Figure 33: LPC11U35/U36/U37 64-pin LQFP package

LPC1224/1225/1226/1227/12D27 PINOUT

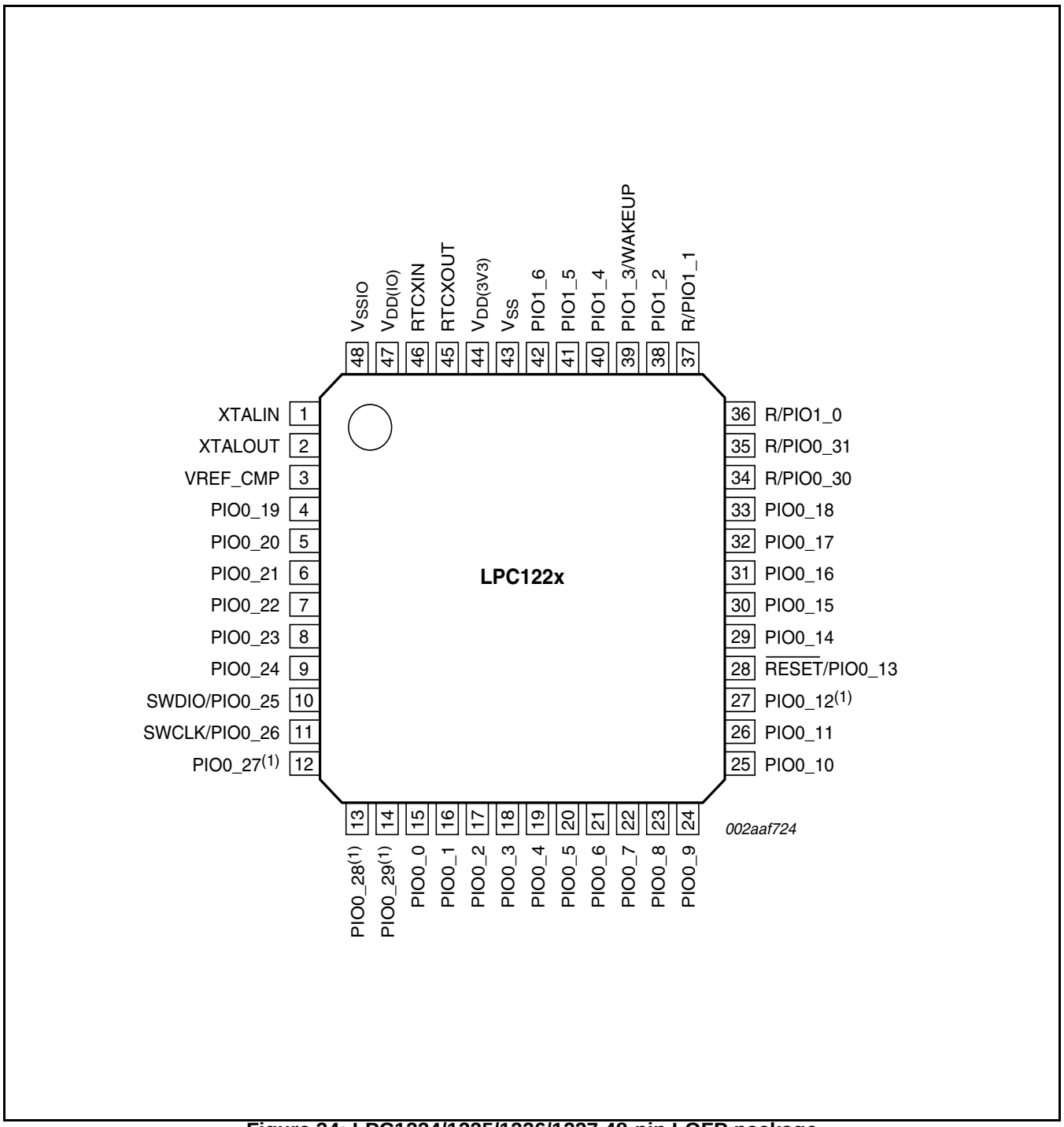


Figure 34: LPC1224/1225/1226/1227 48-pin LQFP package

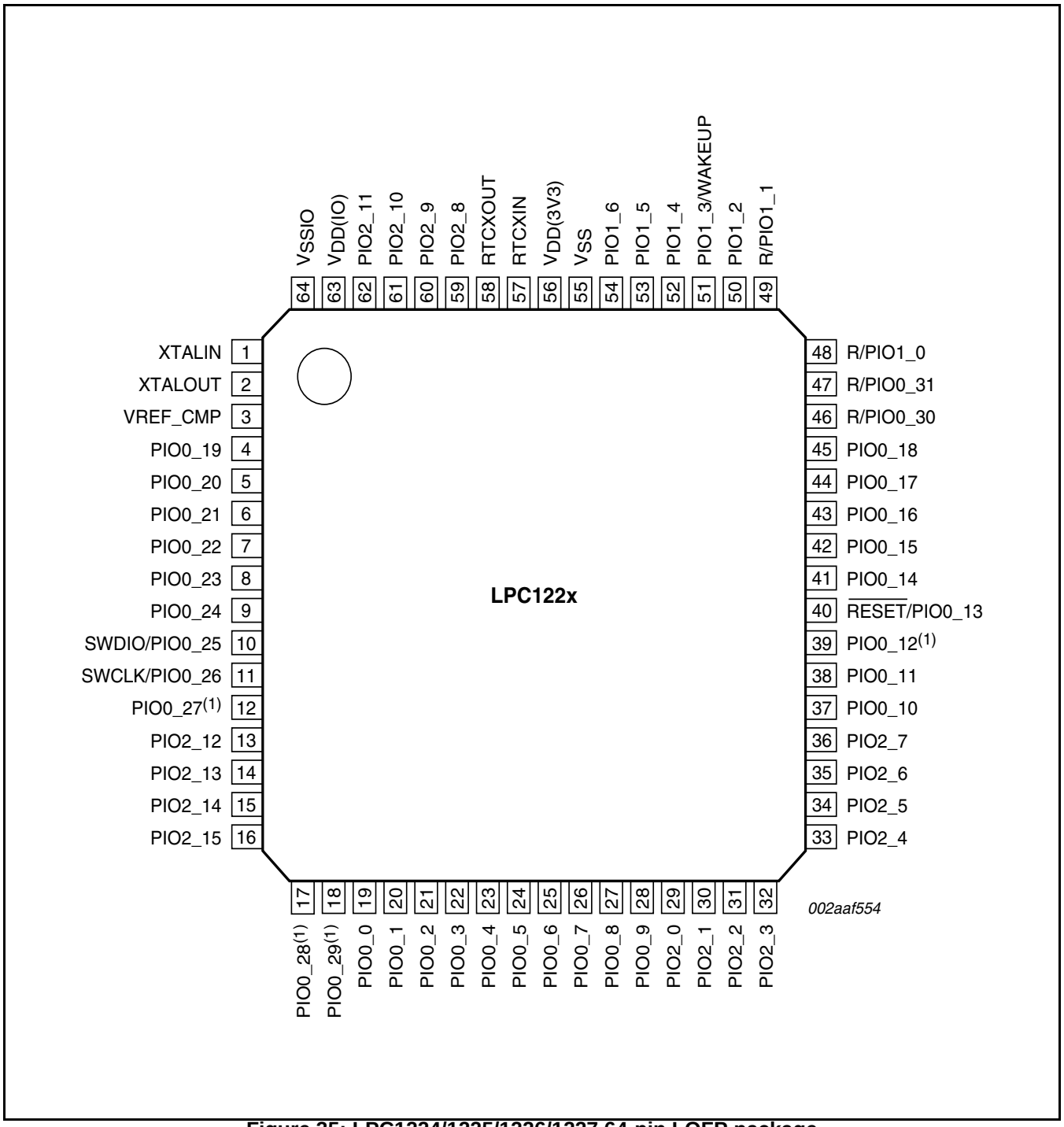


Figure 35: LPC1224/1225/1226/1227 64-pin LQFP package

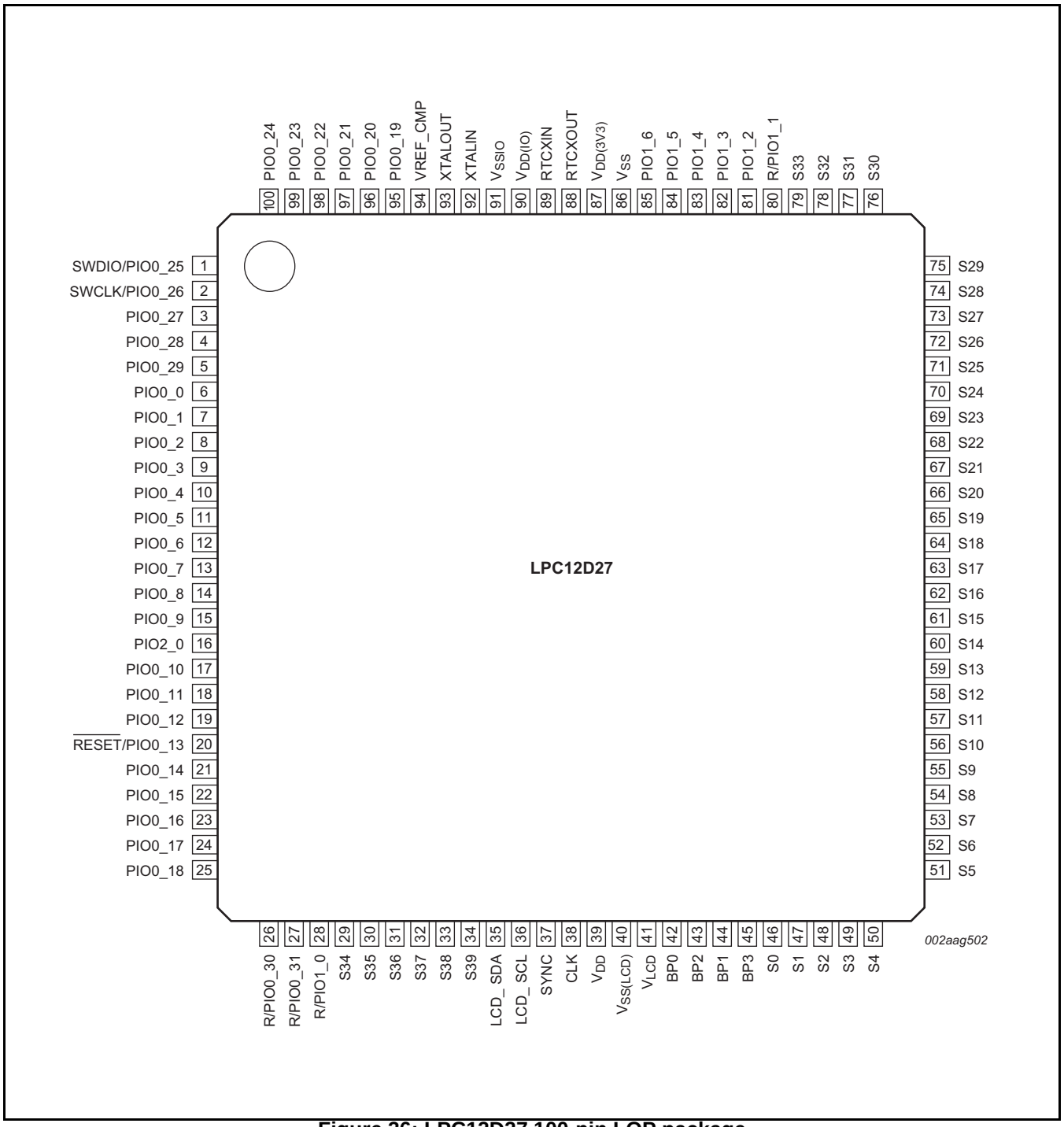


Figure 36: LPC12D27 100-pin LQP package

LPC1311/1313/1342/1343 PINOUT

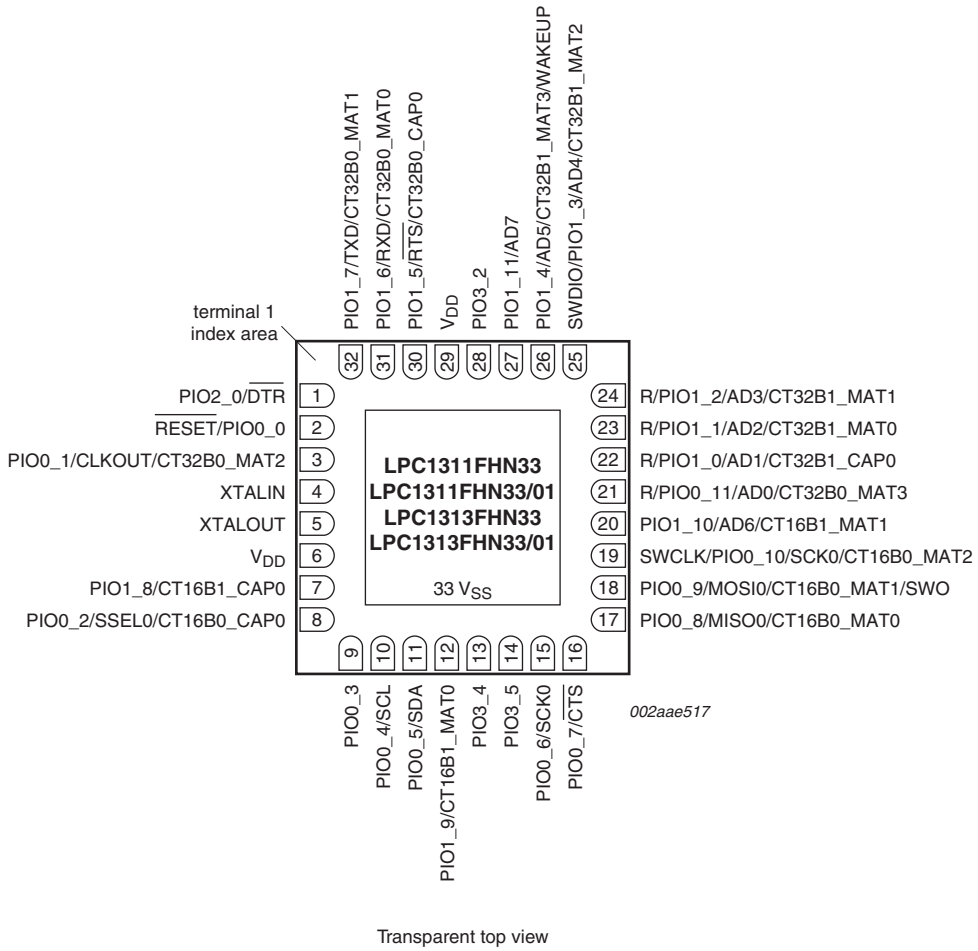


Figure 37: LPC1311/1313 33-pin HVQFN package

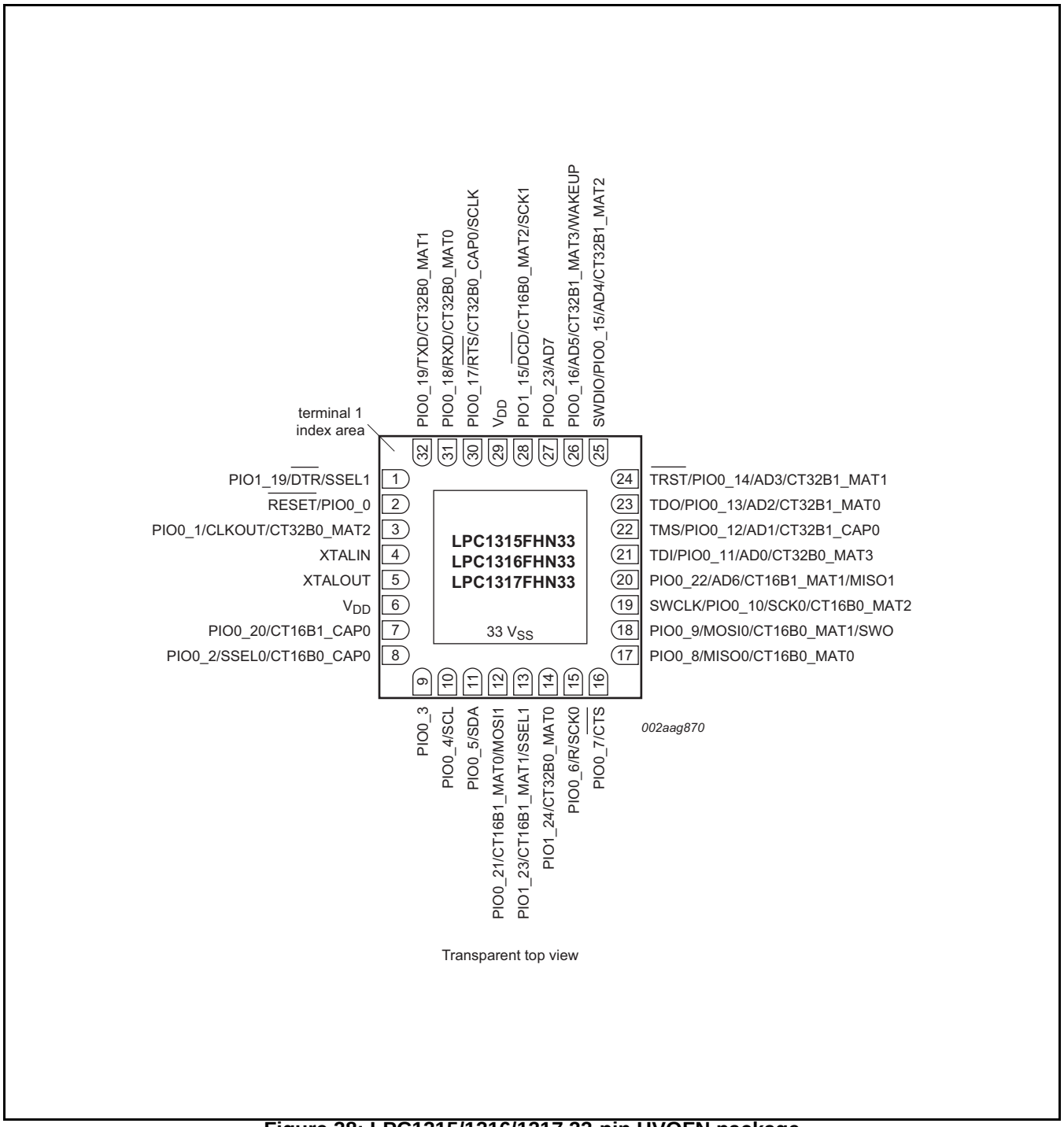


Figure 38: LPC1315/1316/1317 33-pin HVQFN package

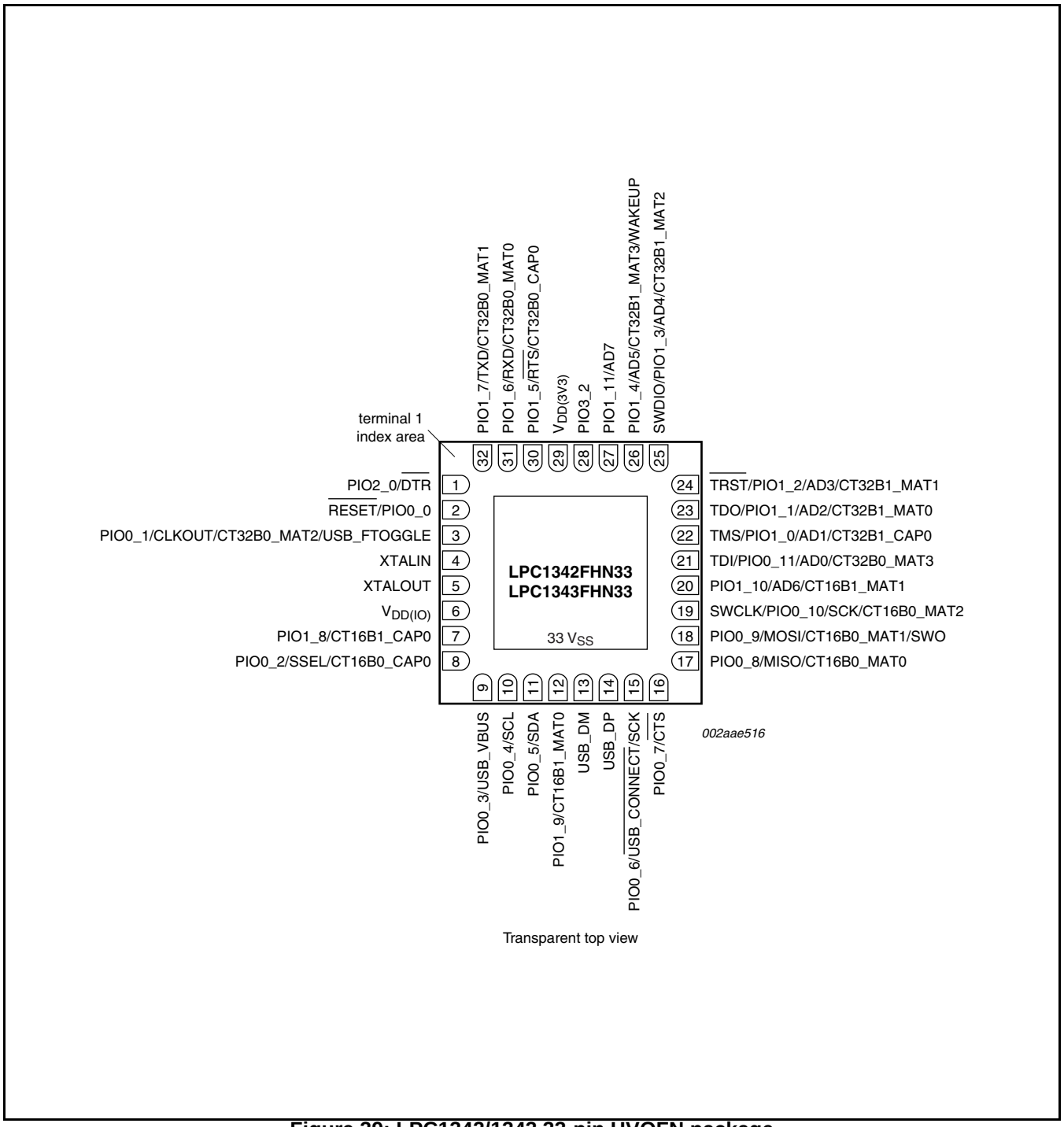


Figure 39: LPC1342/1343 33-pin HVQFN package

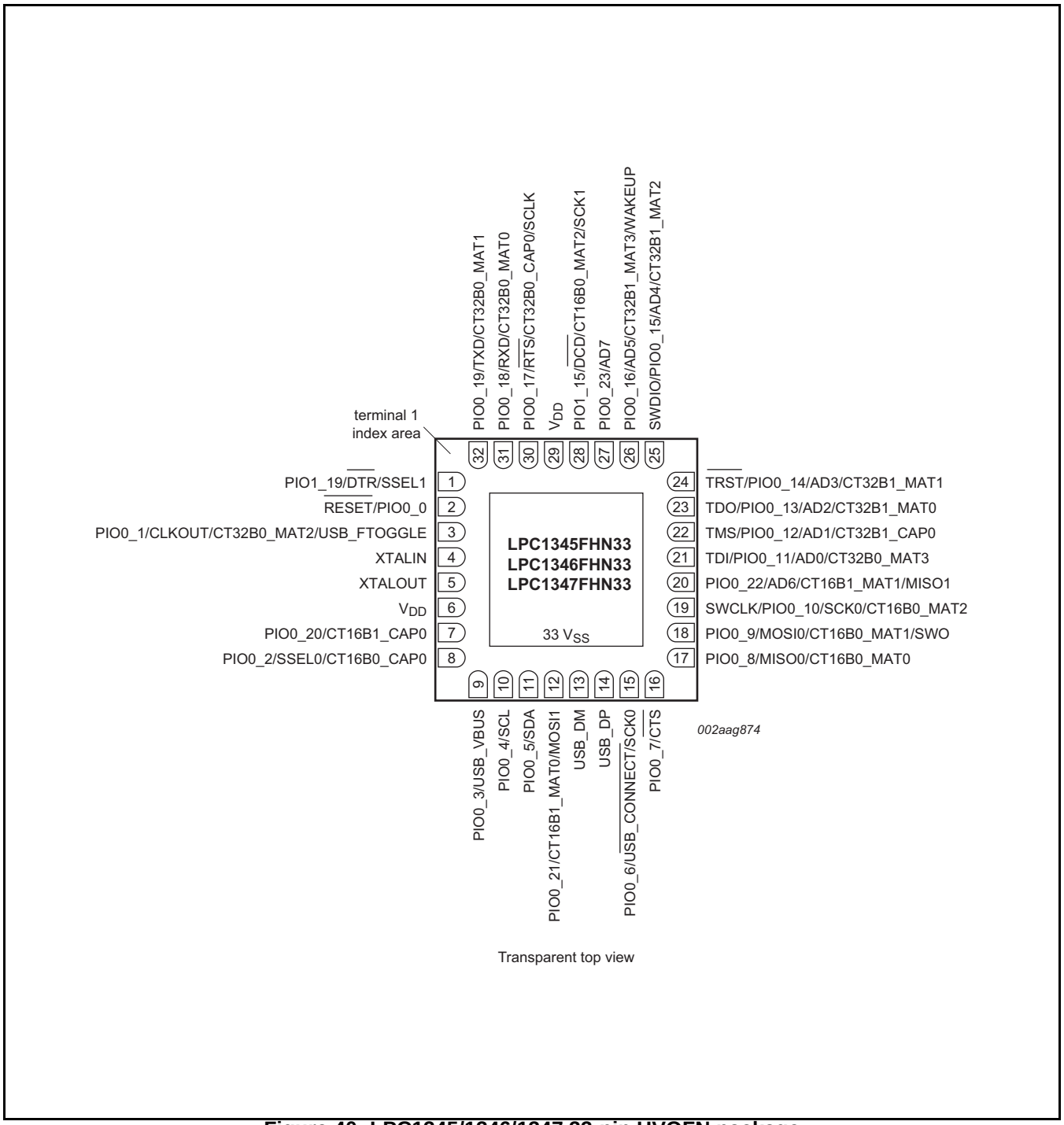


Figure 40: LPC1345/1346/1347 33-pin HVQFN package

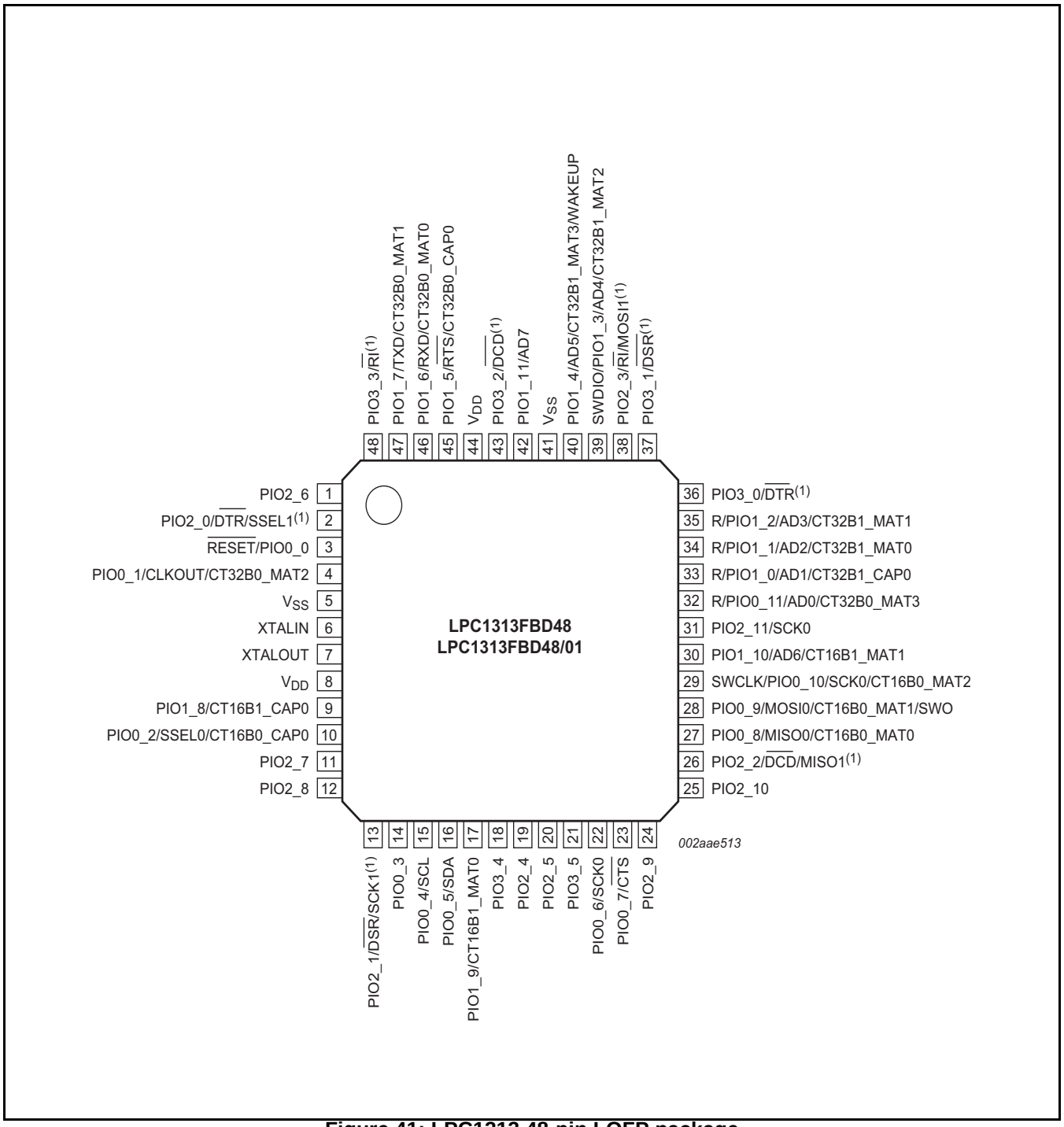


Figure 41: LPC1313 48-pin LQFP package

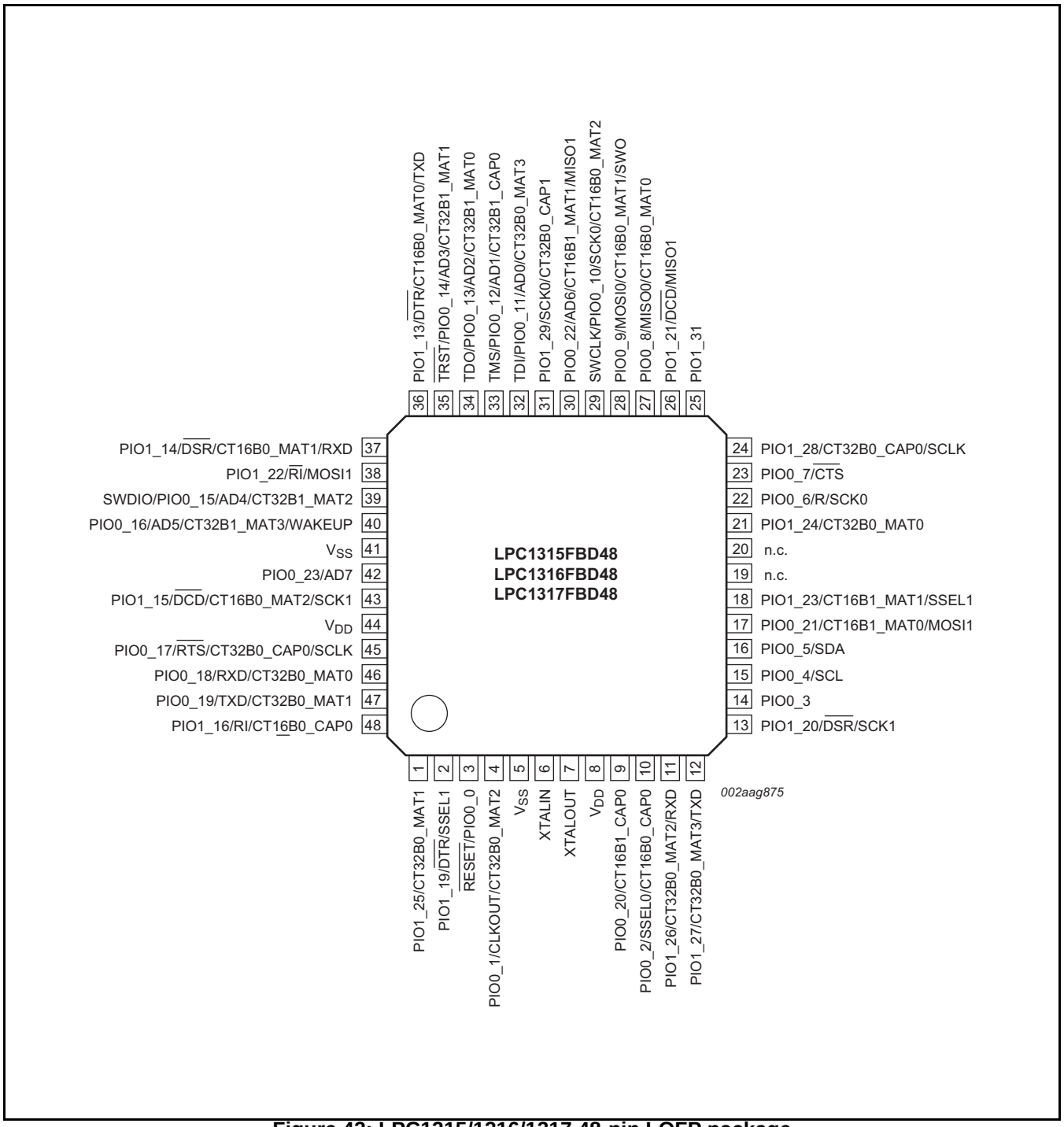


Figure 42: LPC1315/1316/1317 48-pin LQFP package

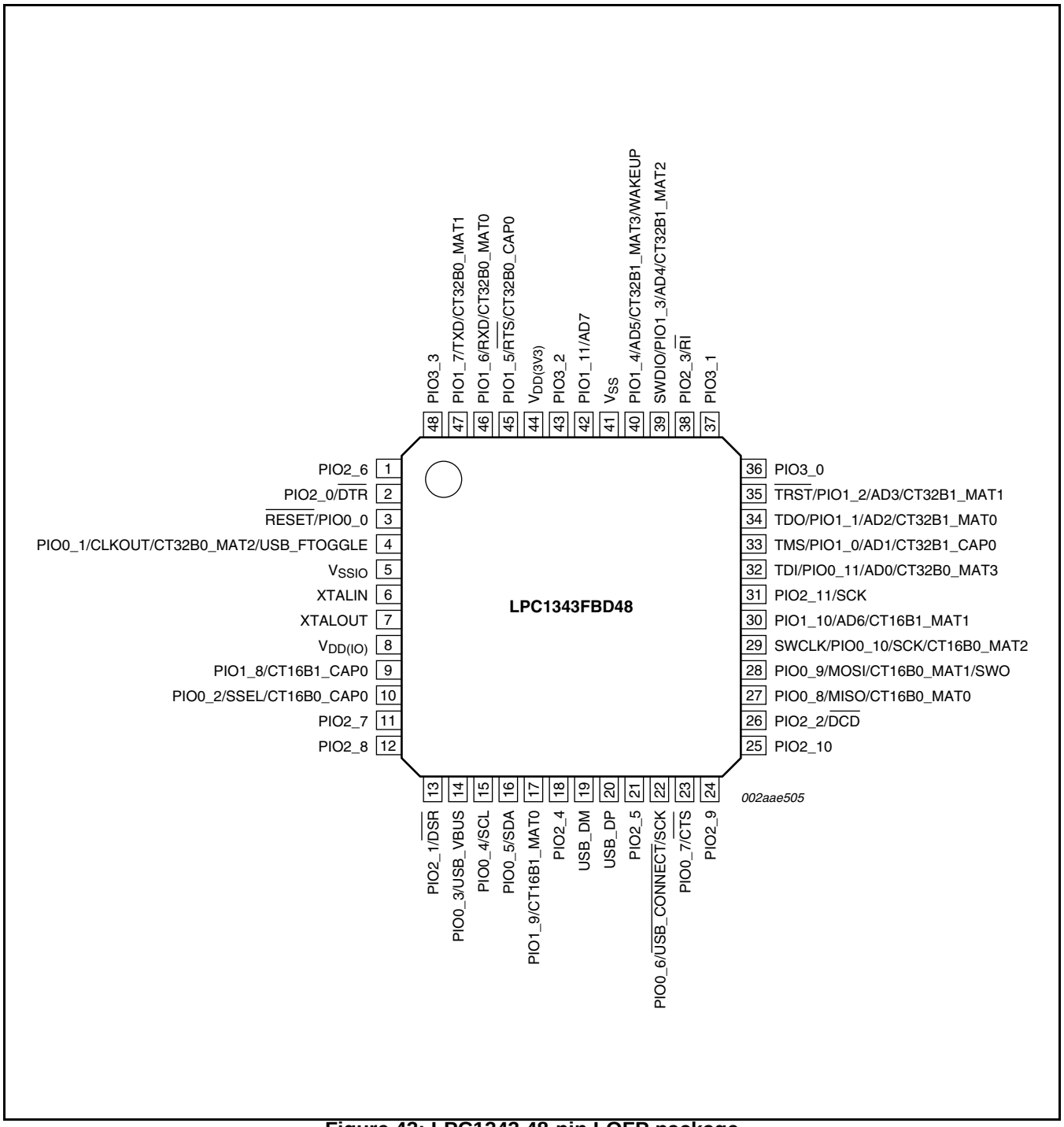


Figure 43: LPC1343 48-pin LQFP package

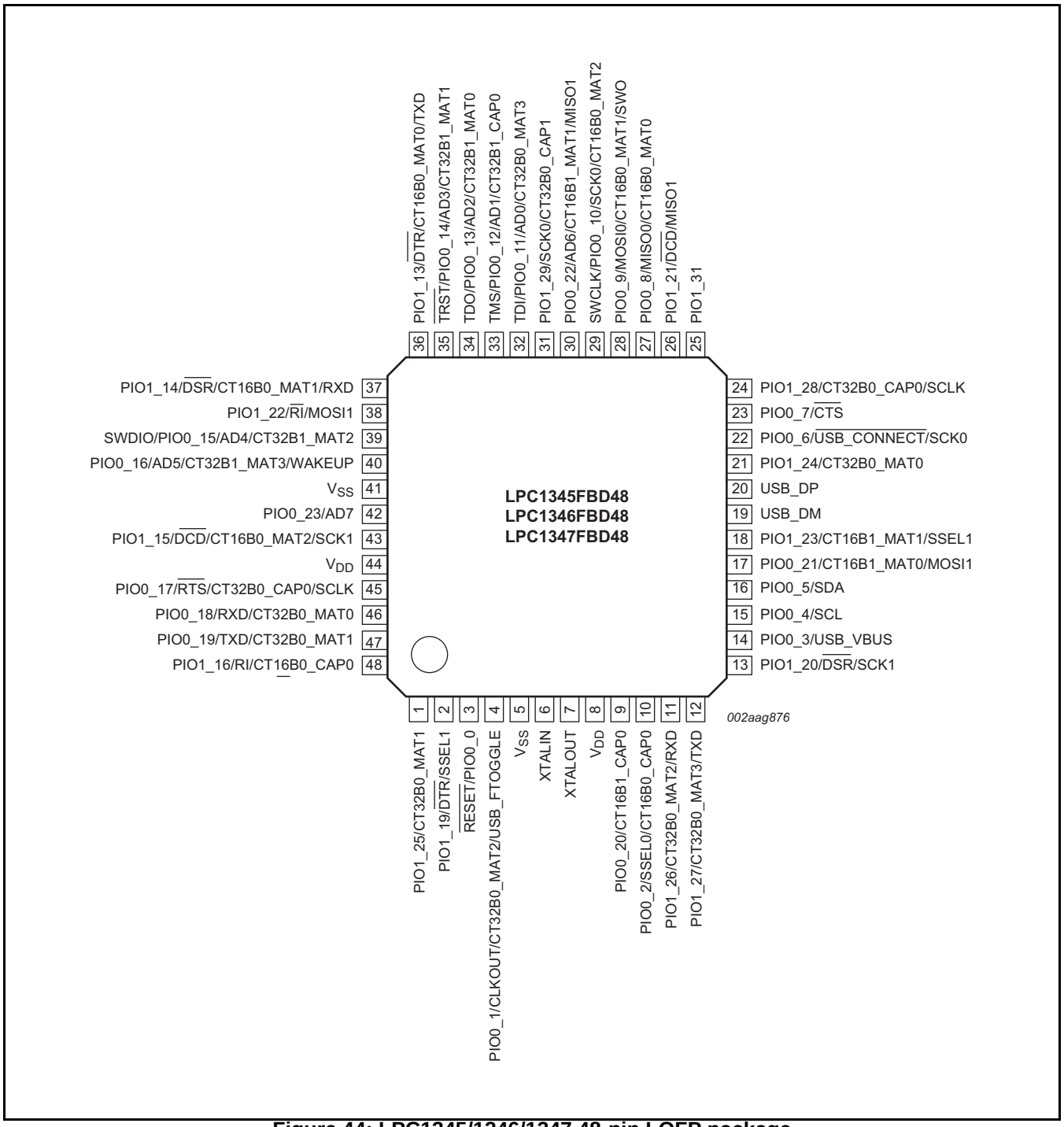


Figure 44: LPC1345/1346/1347 48-pin LQFP package

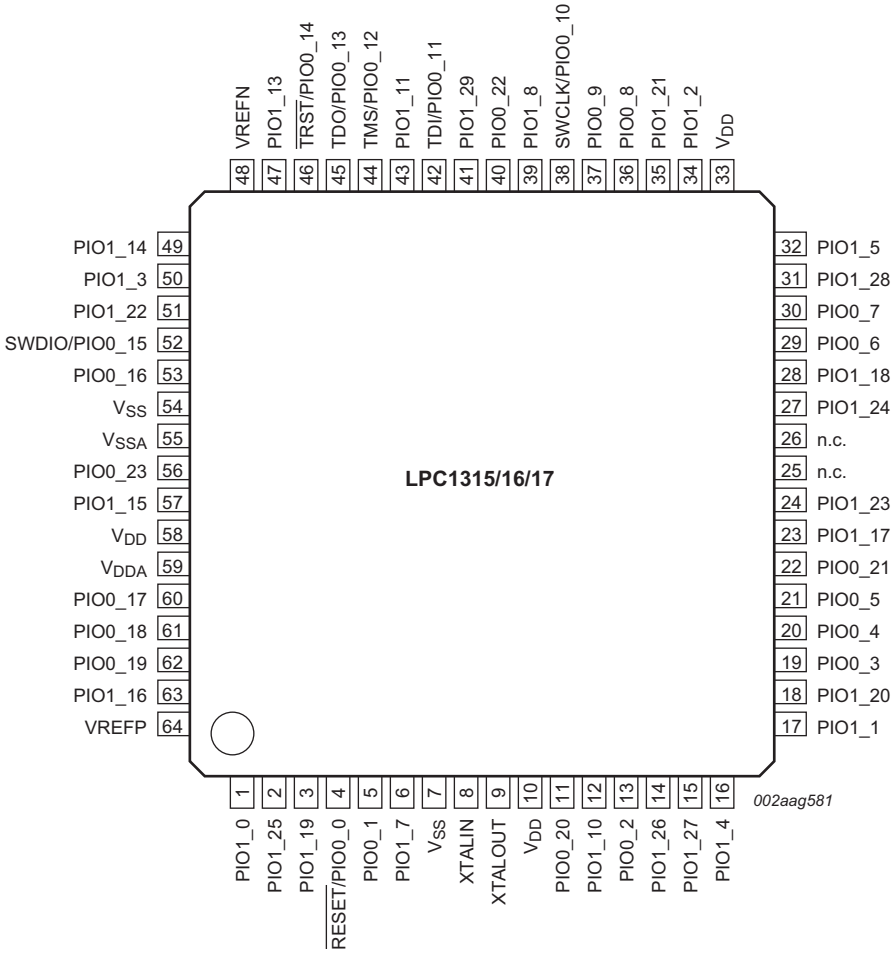


Figure 45: LPC1317 64-pin LQFP package

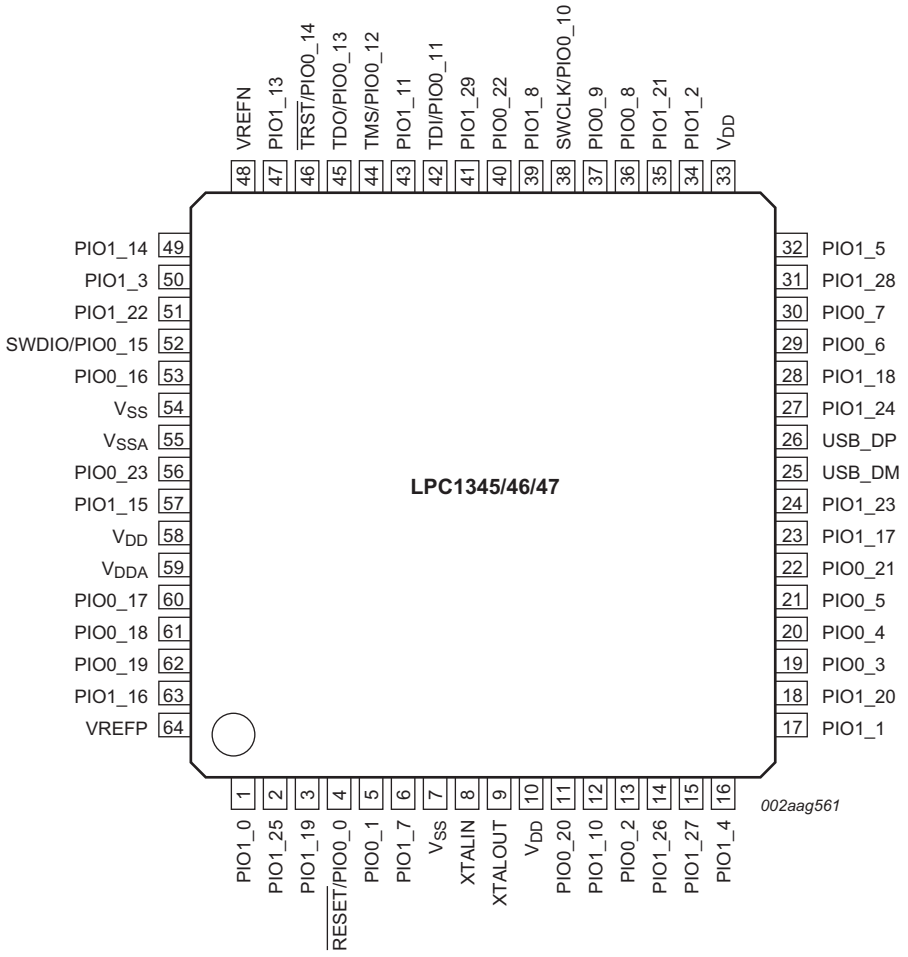


Figure 46: LPC1347 64-pin LQFP package

LPC1751/1752/1754/1756/1758/1759 PINOUT

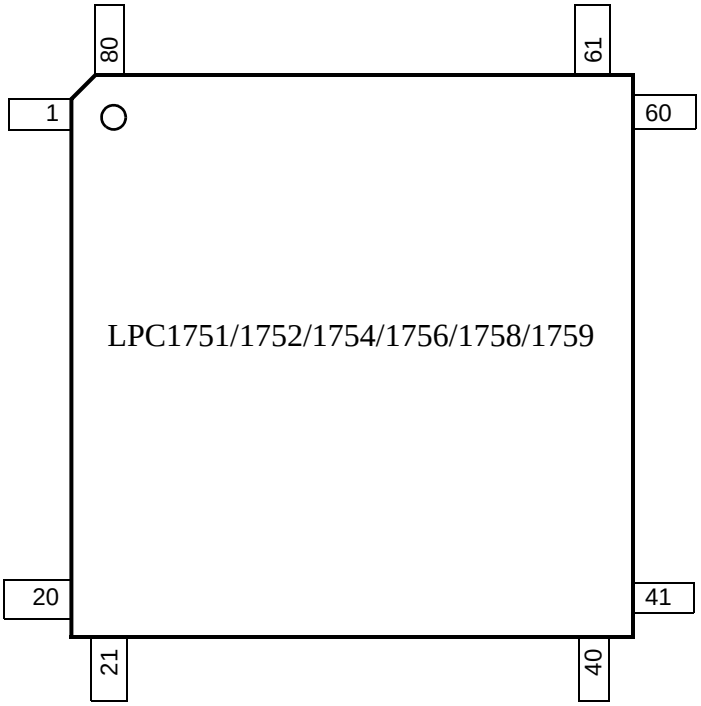


Figure 47: LPC1751/1752/1754/1756/1758/1759 80-pin LQFP package

LPC1763/1764/1765/1766/1767/1768/1769 PINOUT

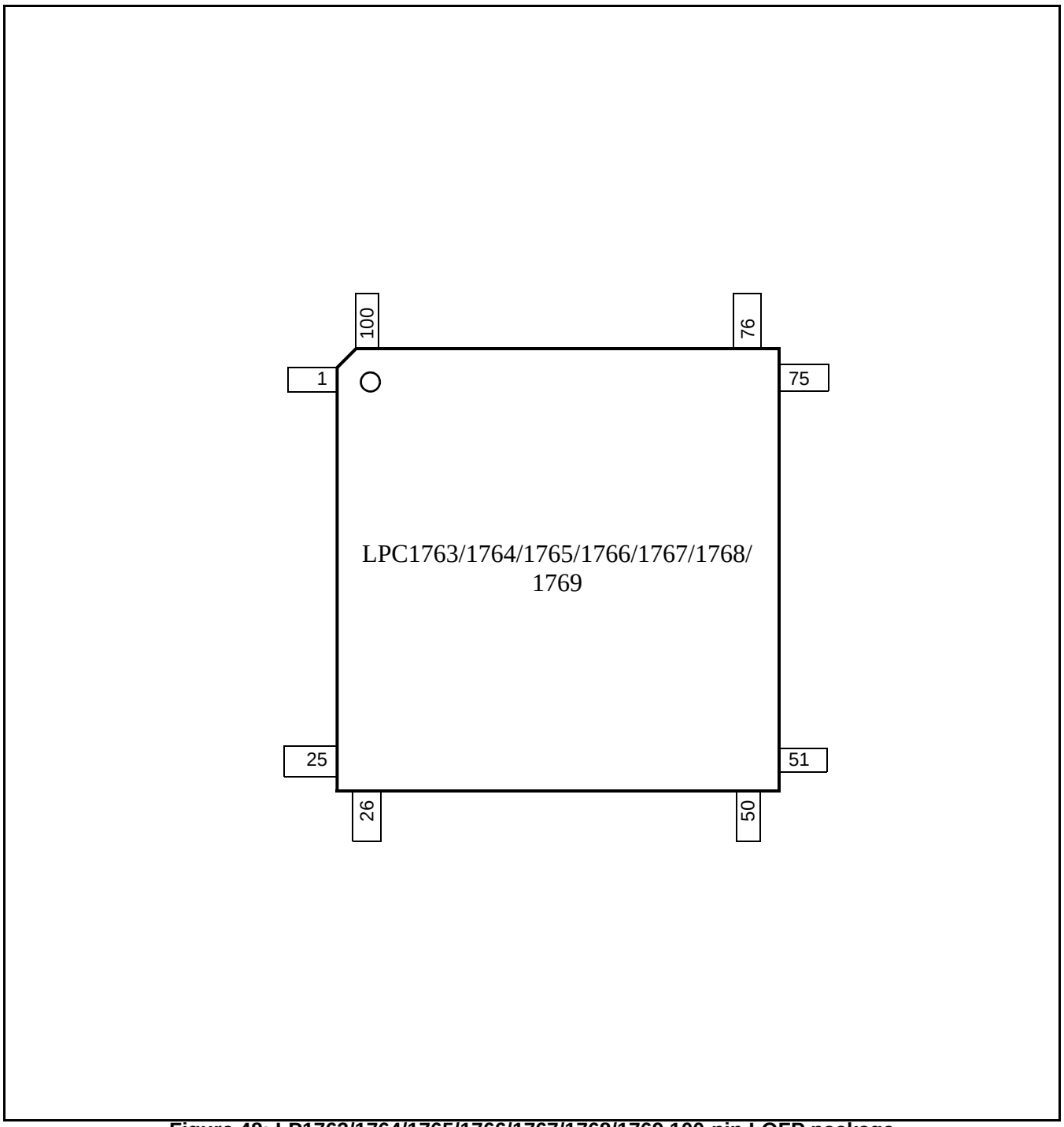


Figure 48: LP1763/1764/1765/1766/1767/1768/1769 100-pin LQFP package

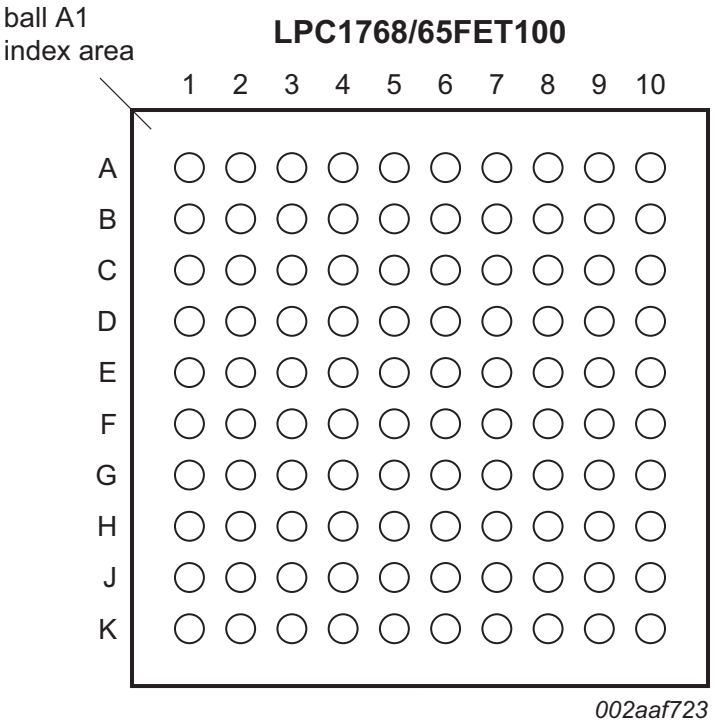


Figure 49: LP1765/1768 100-pin TFBGA package

LPC1774/1778/1788 PINOUT

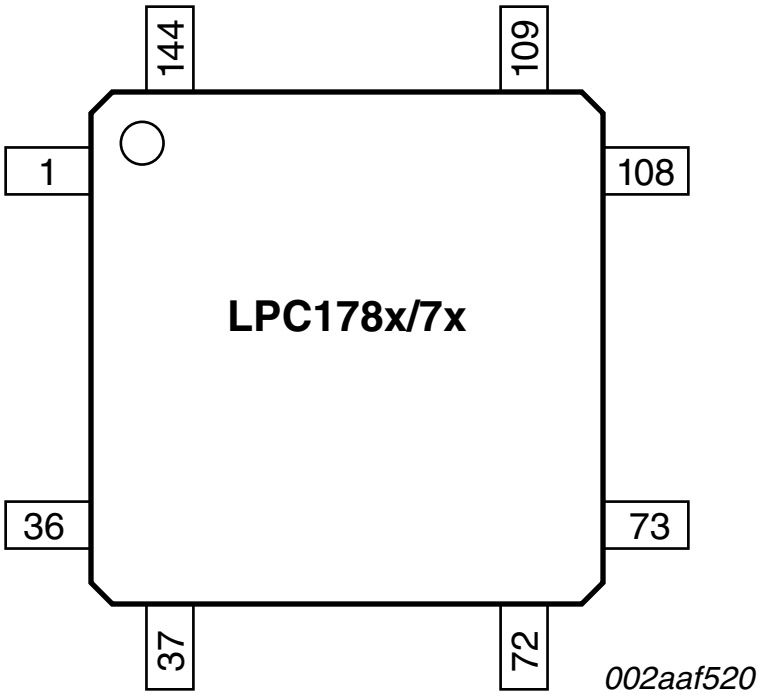


Figure 50: LPC1774/1778/1788 144-pin LQFP package

LPC1776/1778/1788 PINOUT

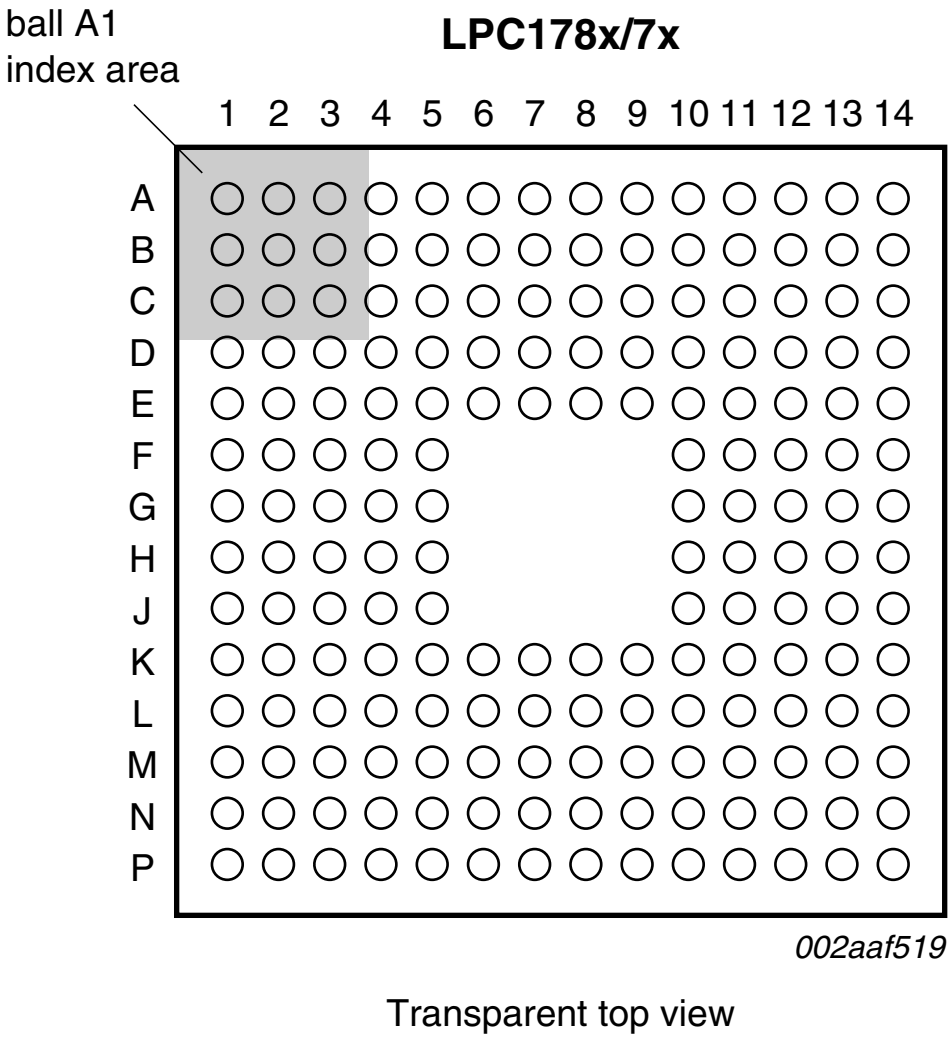


Figure 51: LPC1776/1778/1788 180-pin TFBGA package

LPC1774/1776/1777/1778/1785/1786/1787/1788 PINOUT

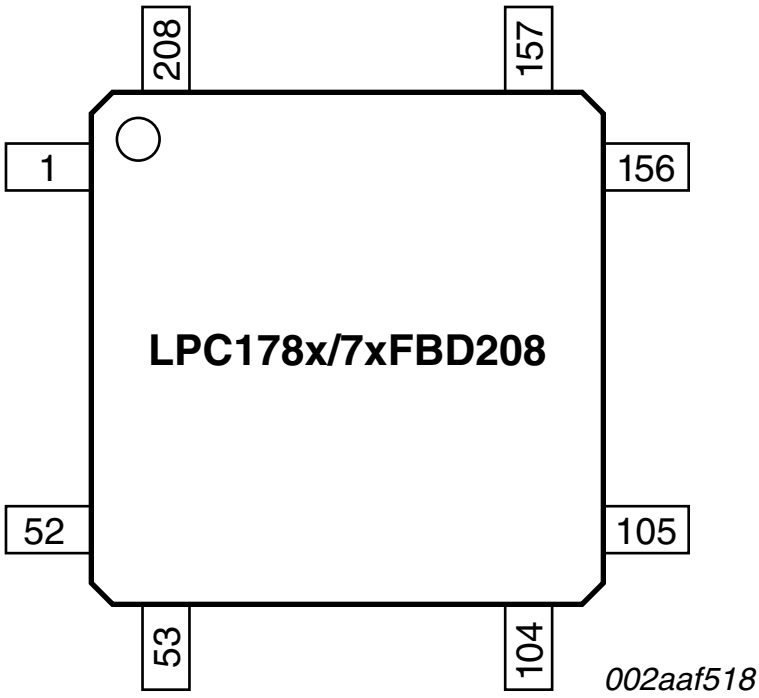


Figure 52: LPC1774/1776/1777/1778/1785/1786/1787/1788 208-pin LQFP Pinout

LPC1778/1788 PINOUT

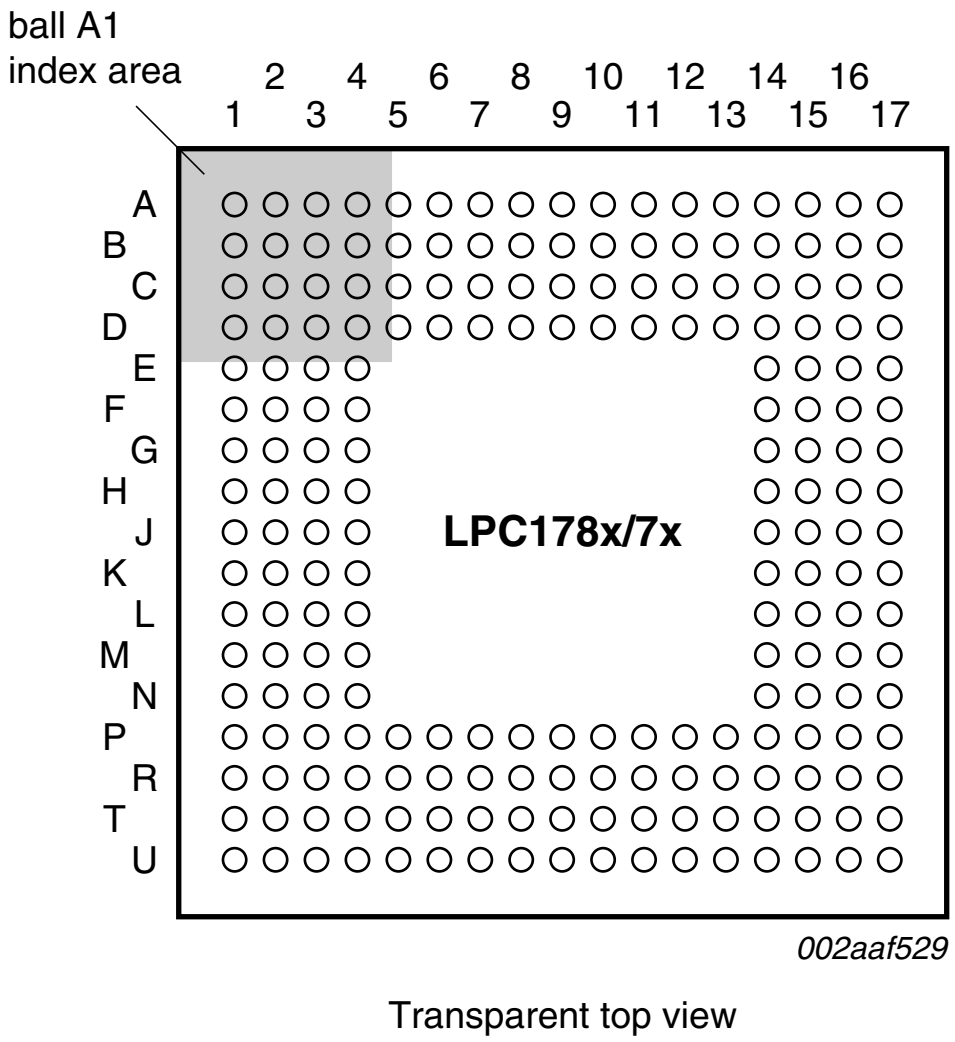


Figure 53: LPC1778/1788 208-pin TFBGA Pinout

LPC1812/1813/1815/1817/1822/1823/1825/1827/1833/1837/1853/1857 PINOUT

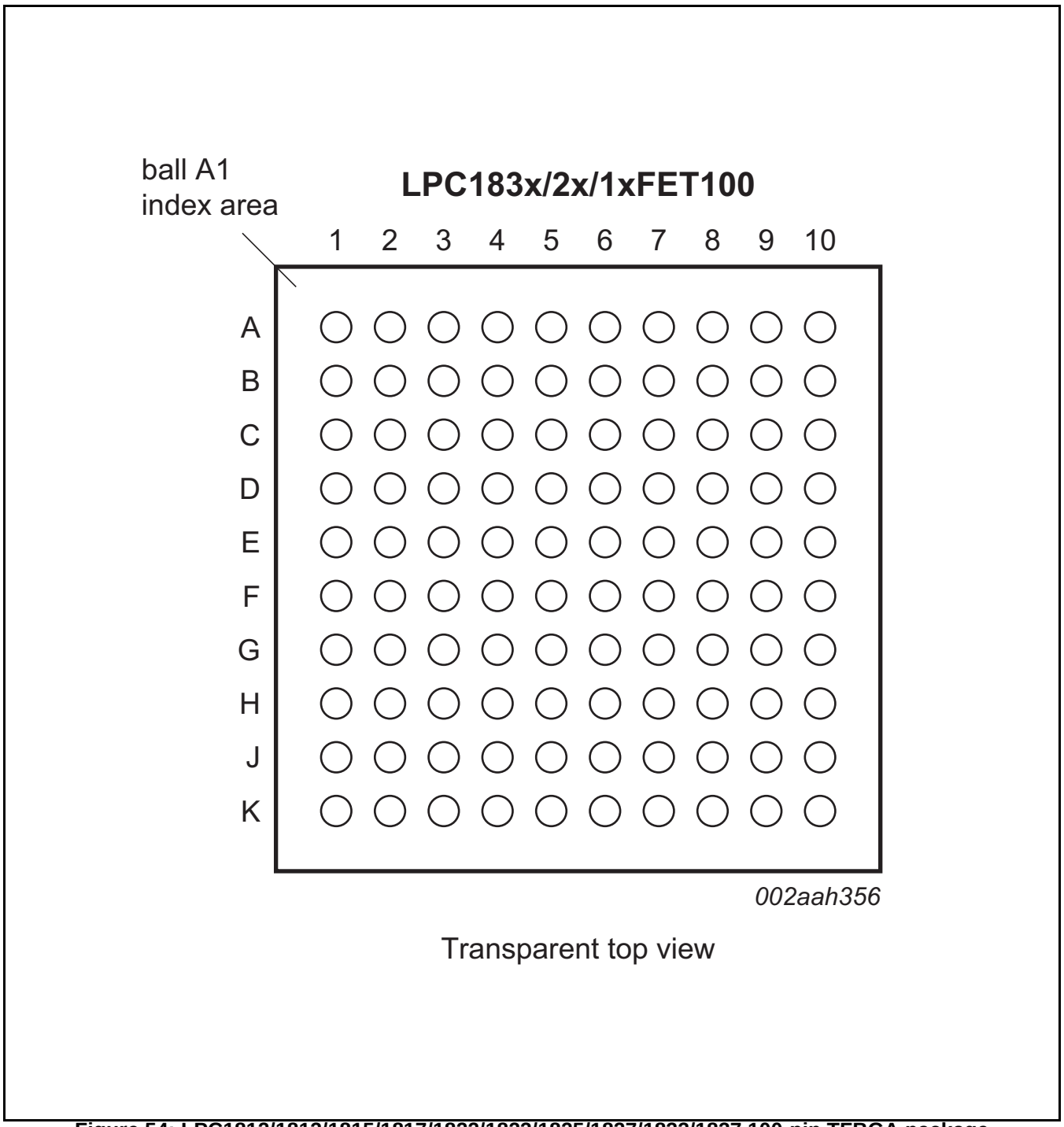


Figure 54: LPC1812/1813/1815/1817/1822/1823/1825/1827/1833/1837 100-pin TFBGA package

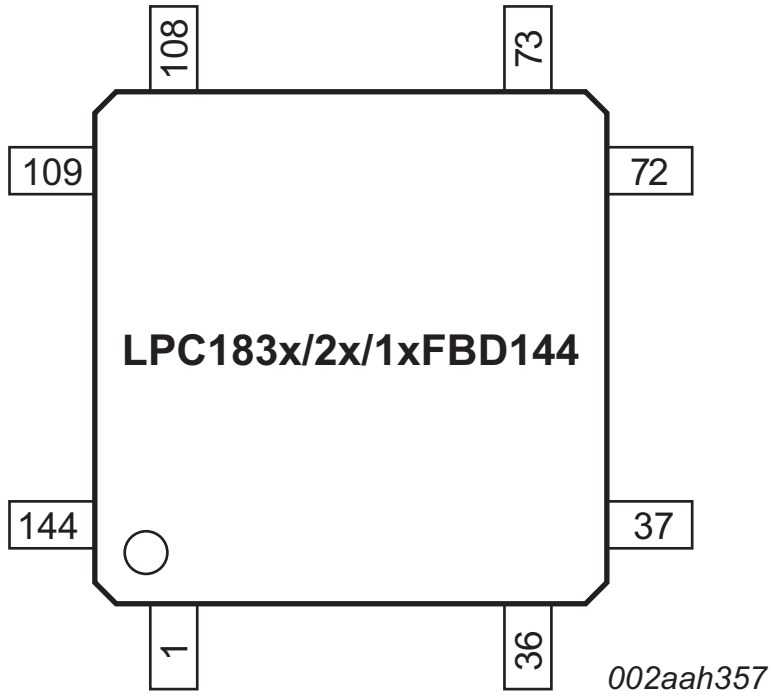


Figure 55: LPC1812/1813/1815/1817/1822/1823/1825/1827/1833/1837 144-pin LQFP package

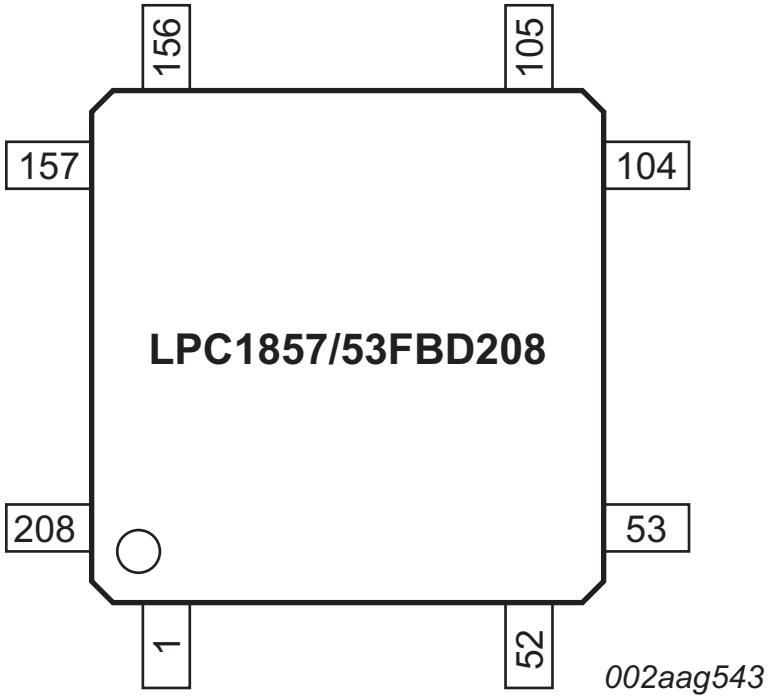


Figure 56: LP1853/1857 208-pin LQFP package

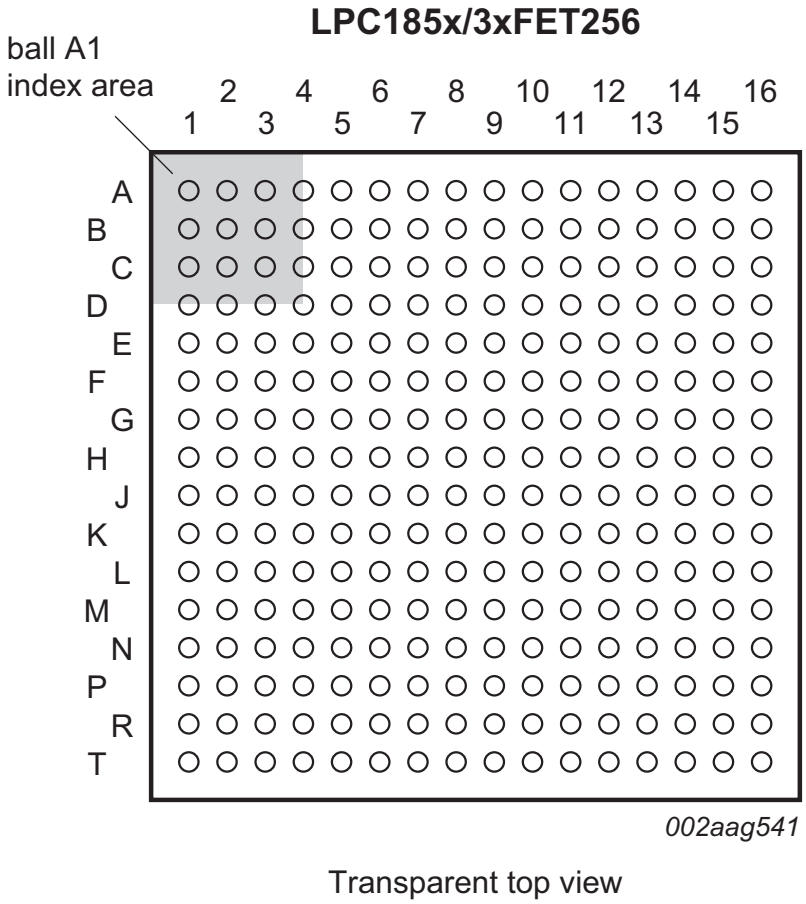


Figure 57: LPC1833/1837/1853/1857 256-pin LBGA package

LPC2101/2102/2103 PINOUT

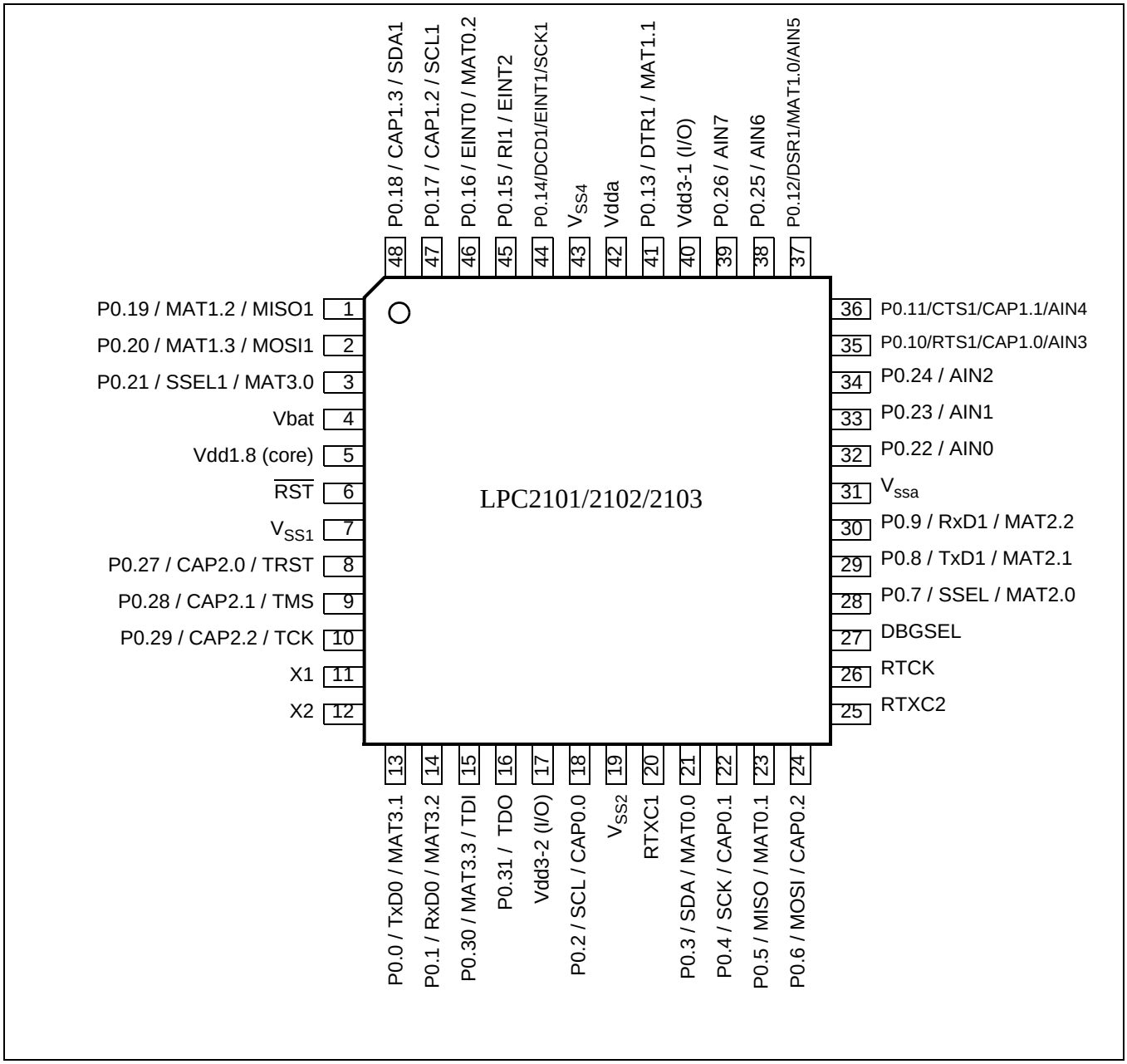


Figure 58: Pin configuration for LPC2101/2102/2103

LPC2104/2105/2106 PINOUT

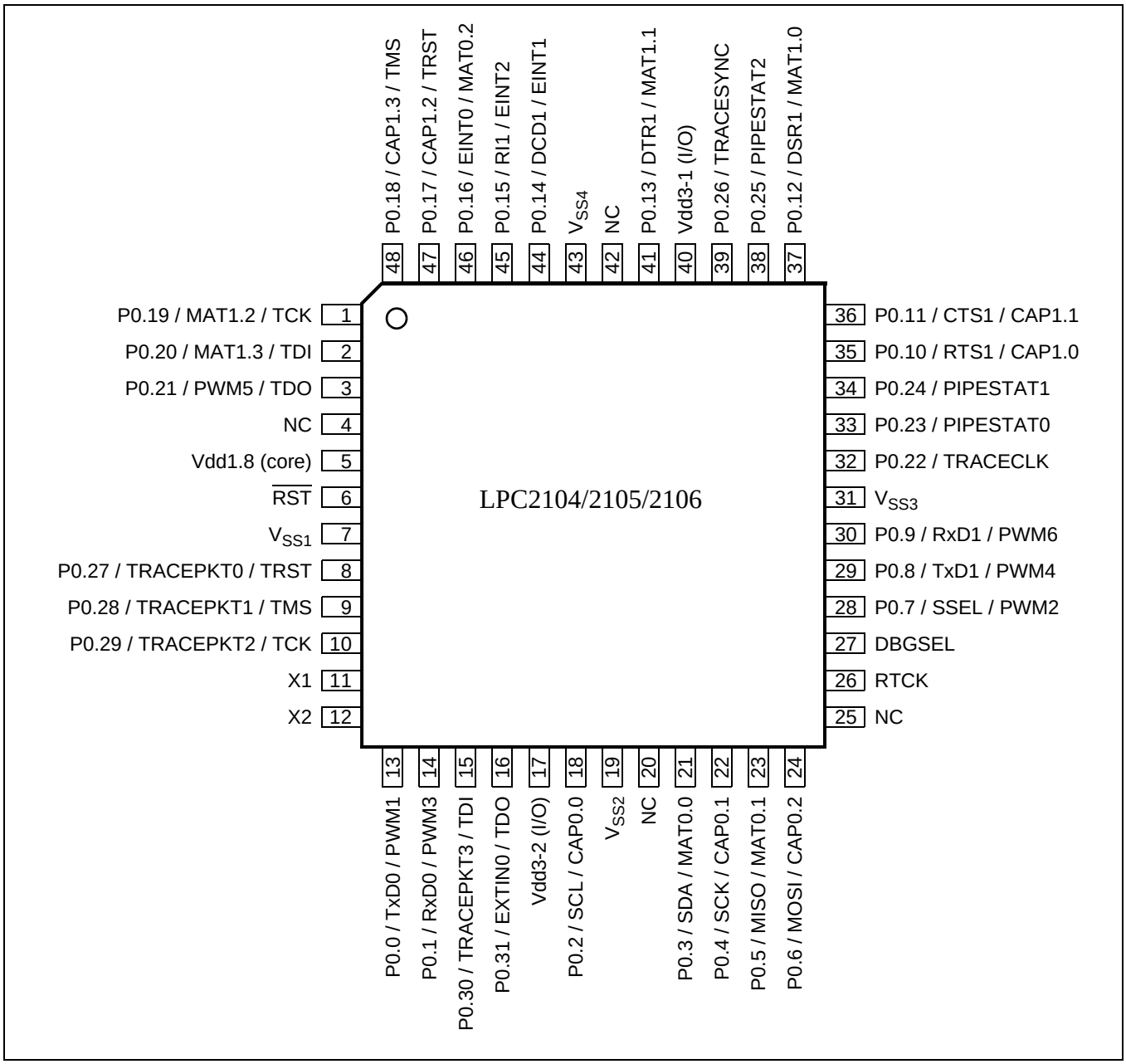


Figure 59: Pin configuration for LPC2104/2105/2106

LPC2114/2124 PINOUT

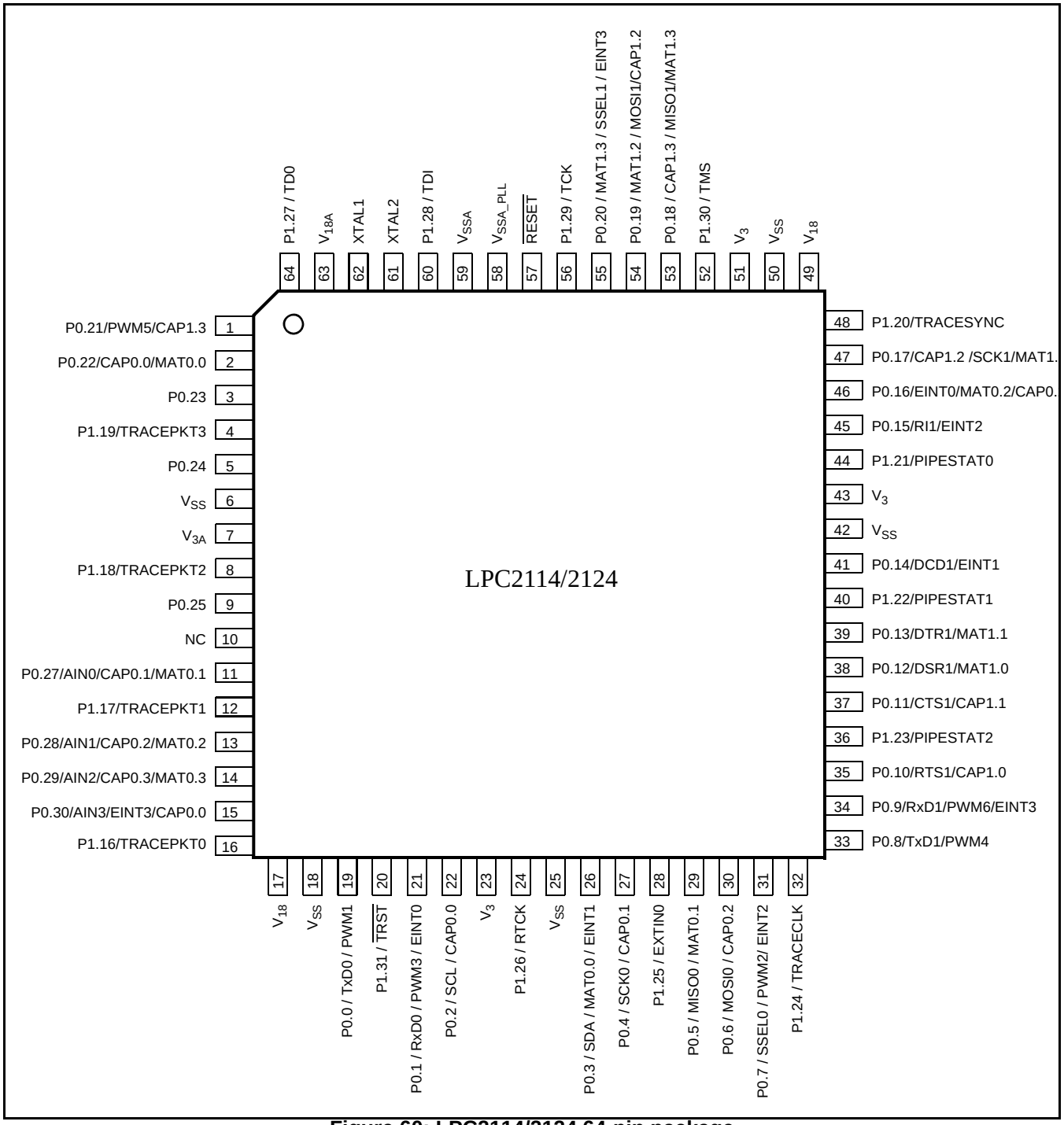


Figure 60: LPC2114/2124 64-pin package

LPC2109/2119/2129/2194 PINOUT

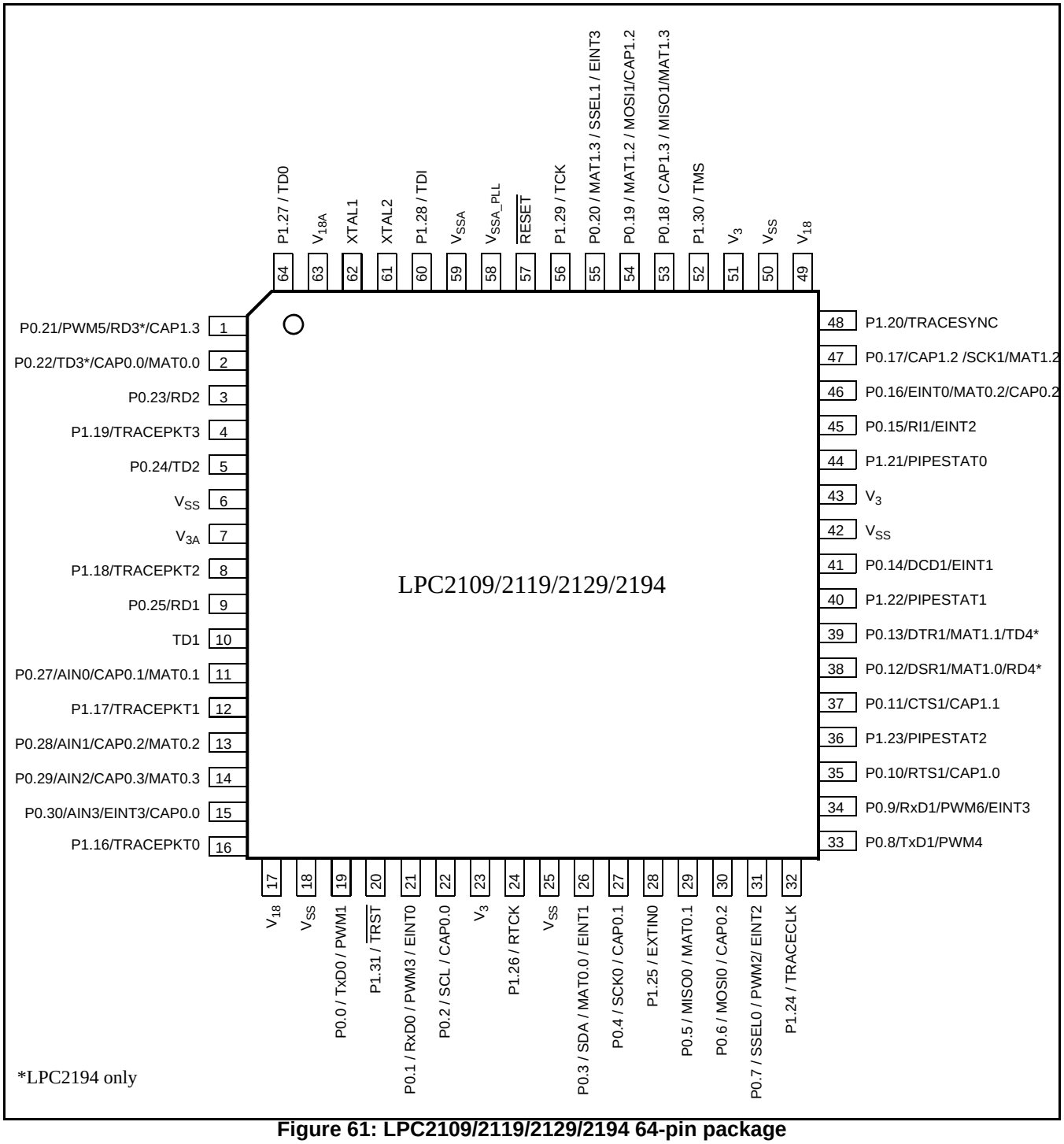


Figure 61: LPC2109/2119/2129/2194 64-pin package

LPC2131/2132/2134/2136/2138 PINOUT

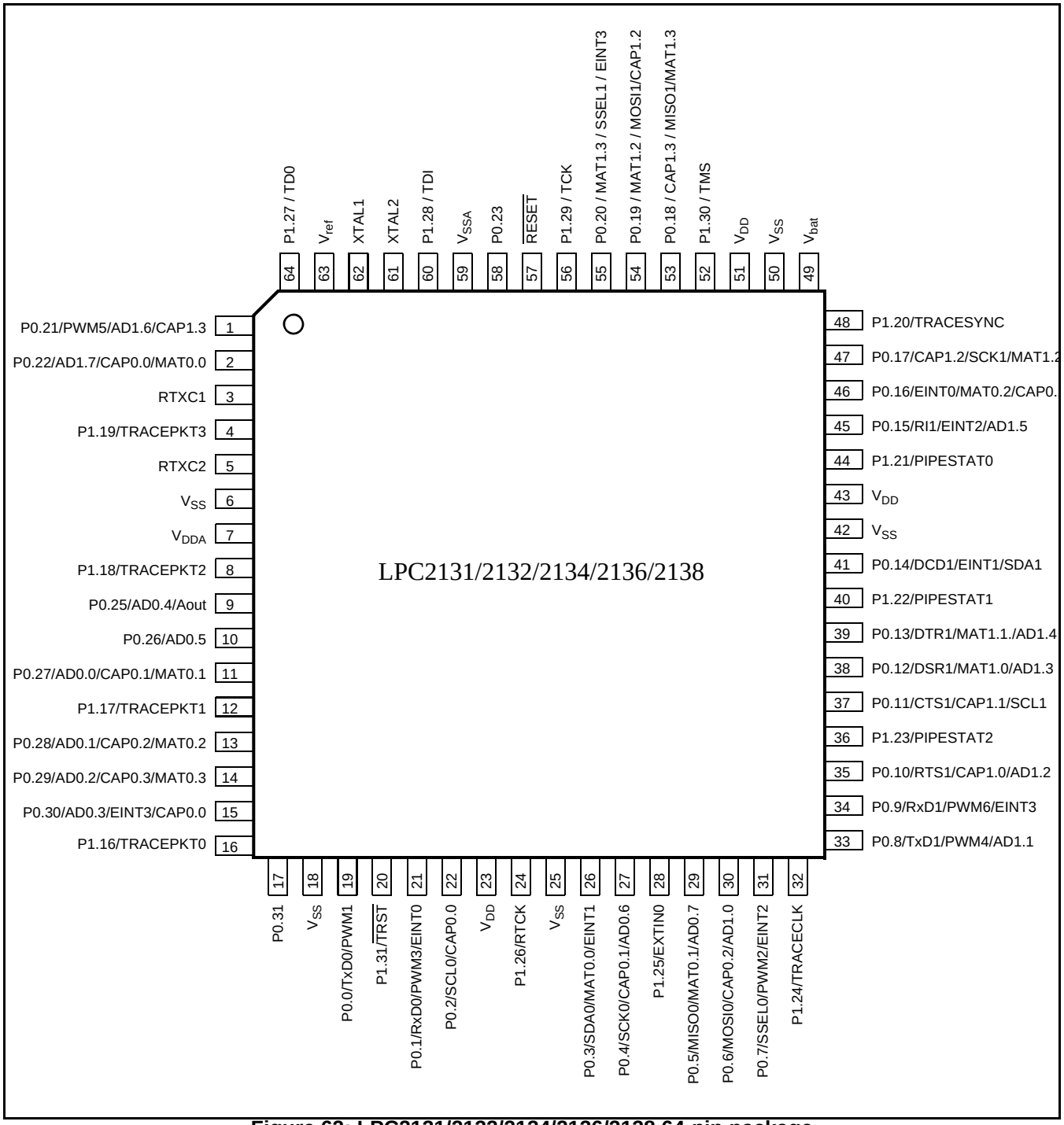


Figure 62: LPC2131/2132/2134/2136/2138 64-pin package

LPC2141/2142/2144/2146/2148 PINOUT

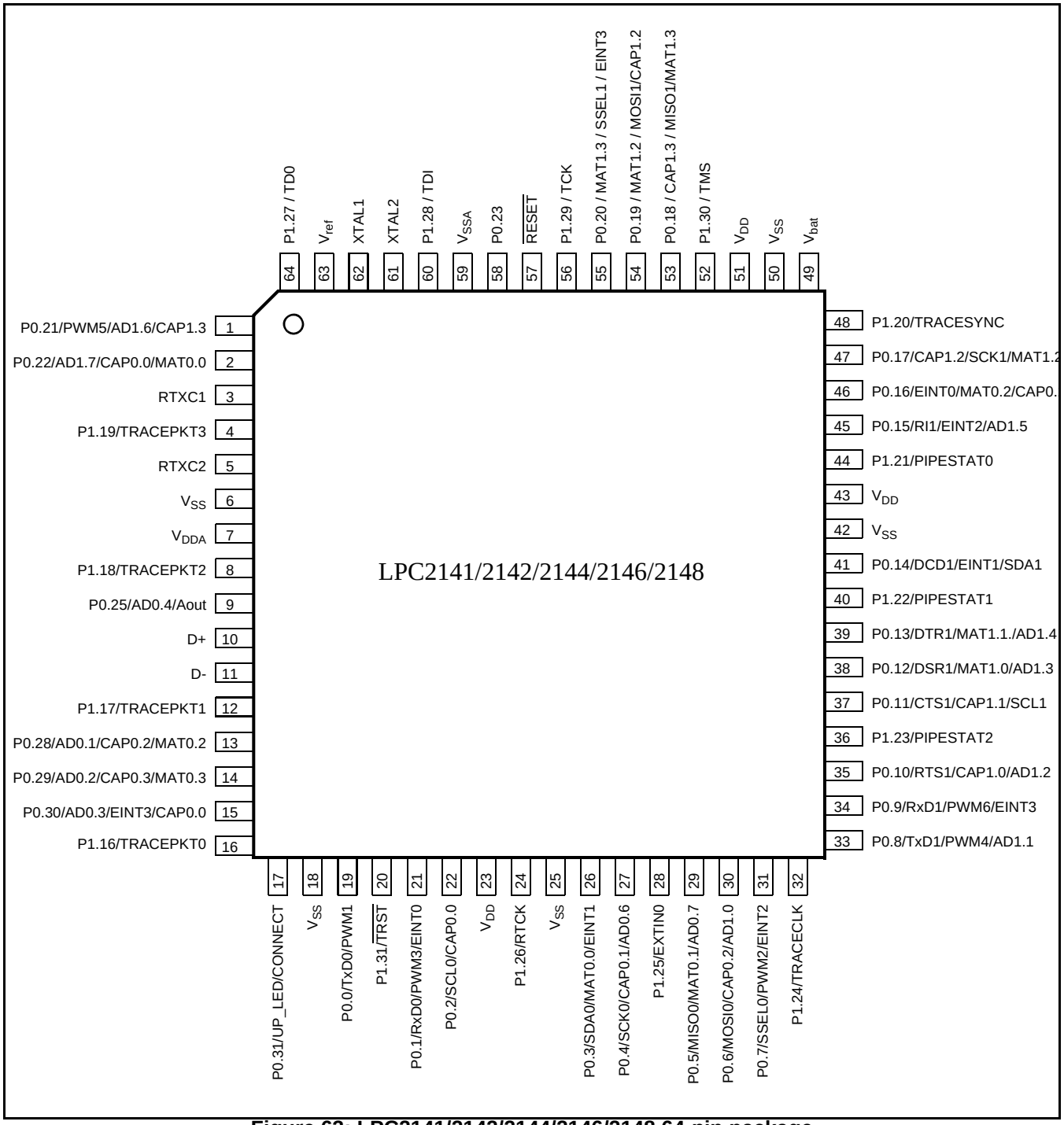


Figure 63: LPC2141/2142/2144/2146/2148 64-pin package

LPC2212/2214 PINOUT

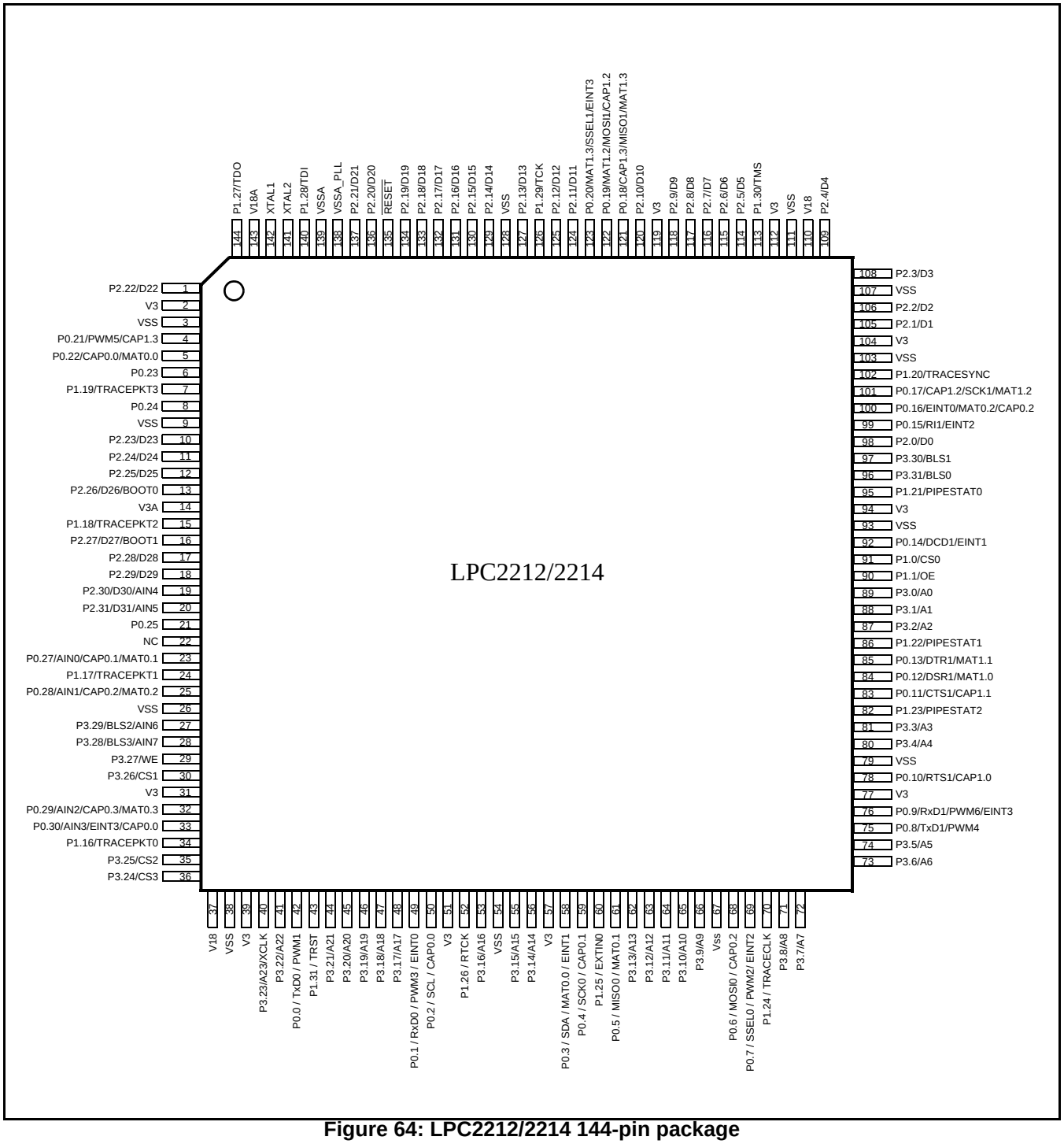
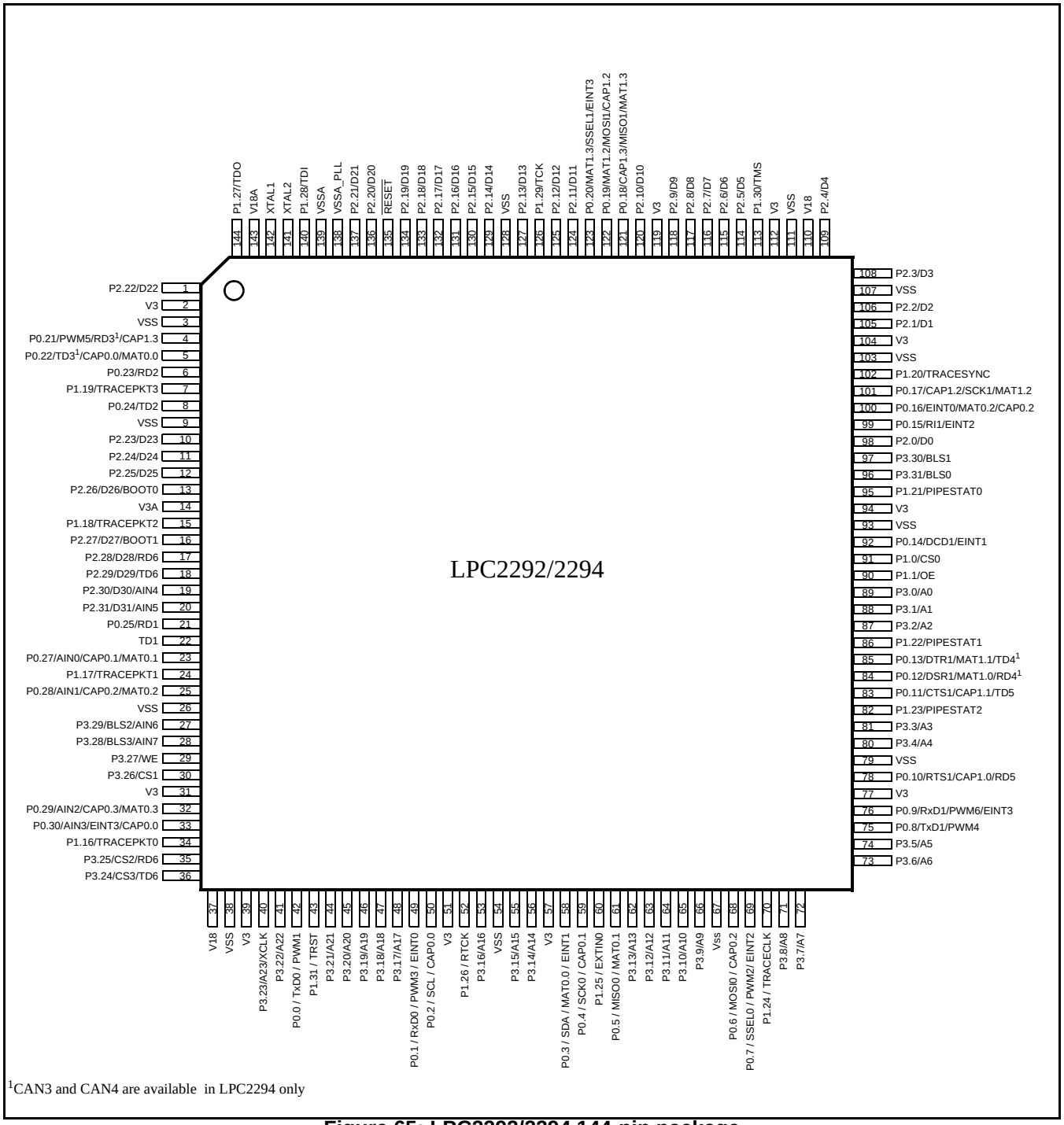


Figure 64: LPC2212/2214 144-pin package

LPC2292/2294 PINOUT



[1]CAN3 and CAN4 are available in LPC2294 only

Figure 65: LPC2292/2294 144-pin package

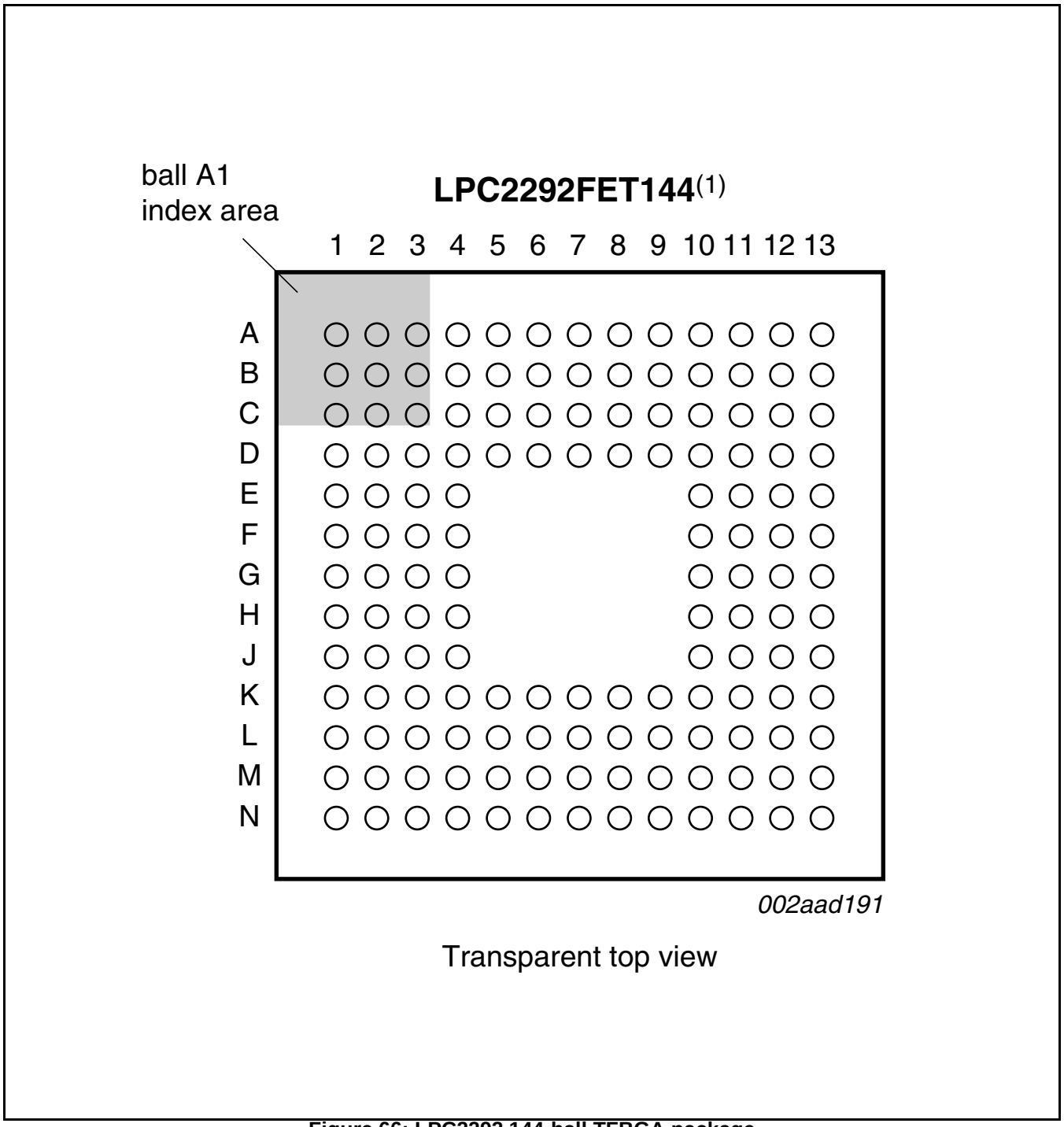


Figure 66: LPC2292 144-ball TFBGA package

LPC2361/2362/2364/2365/2366/2367/2368/2387 PINOUT

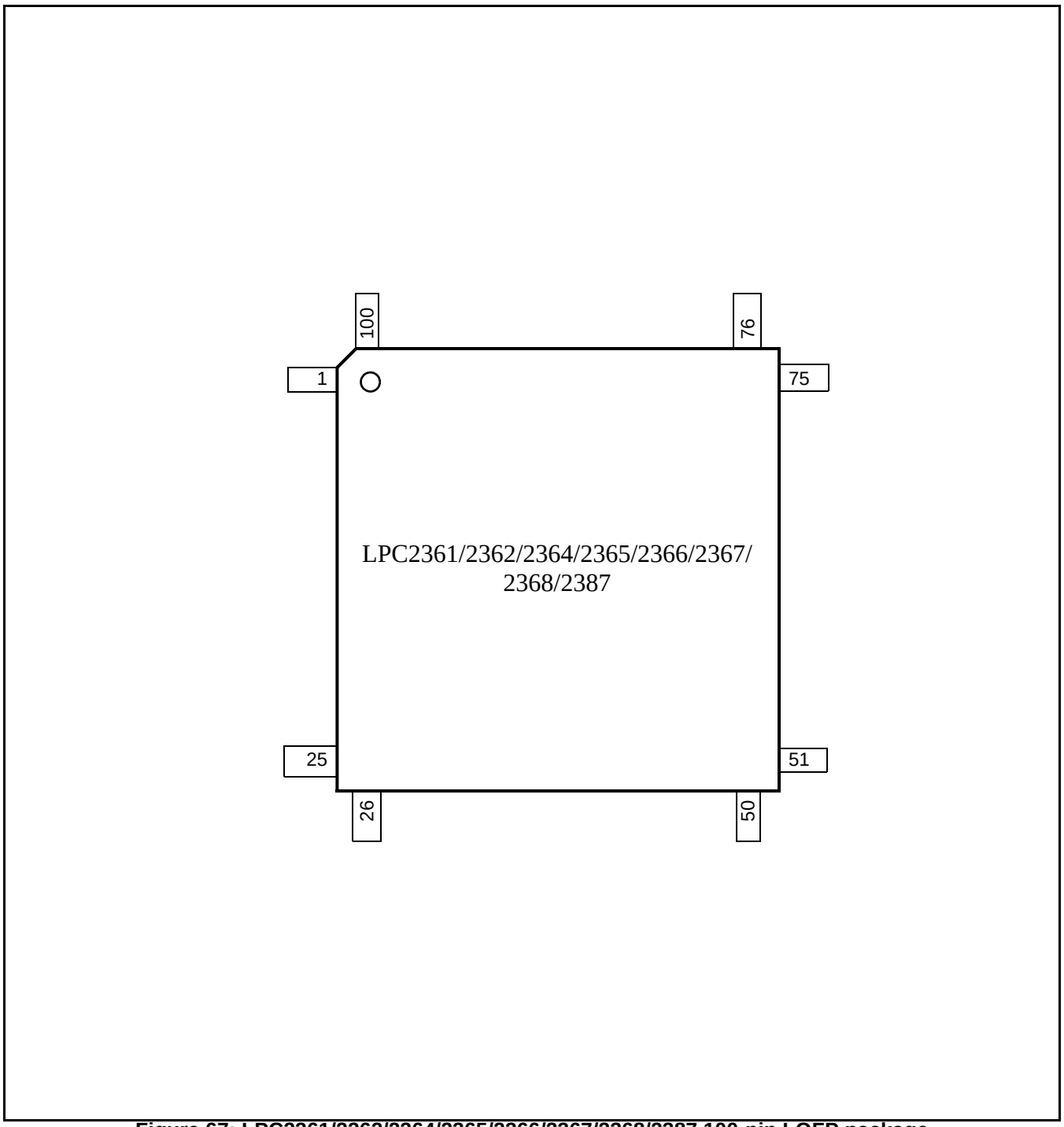


Figure 67: LPC2361/2362/2364/2365/2366/2367/2368/2387 100-pin LQFP package

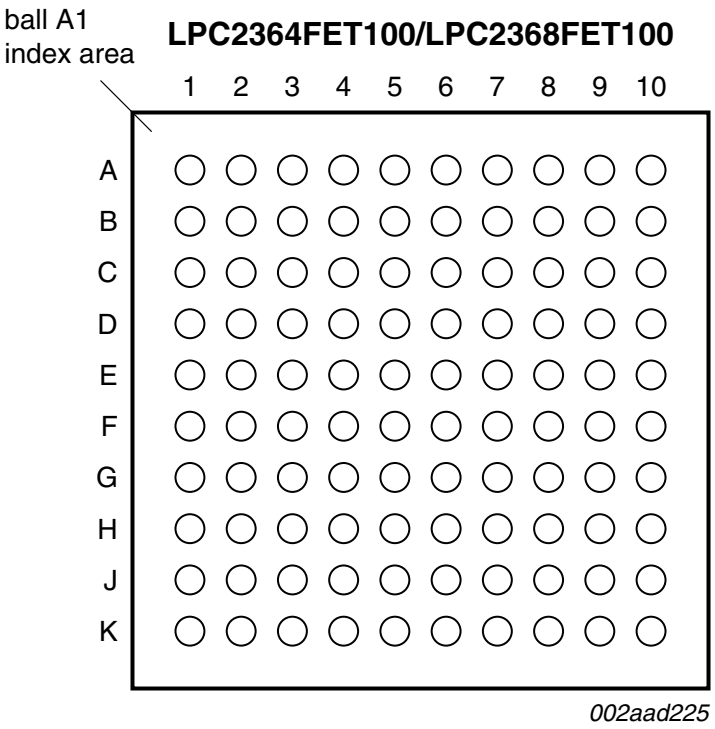


Figure 68: LPC2364/2368 100-ball TFBGA package

LPC2377/2378/2388 PINOUT

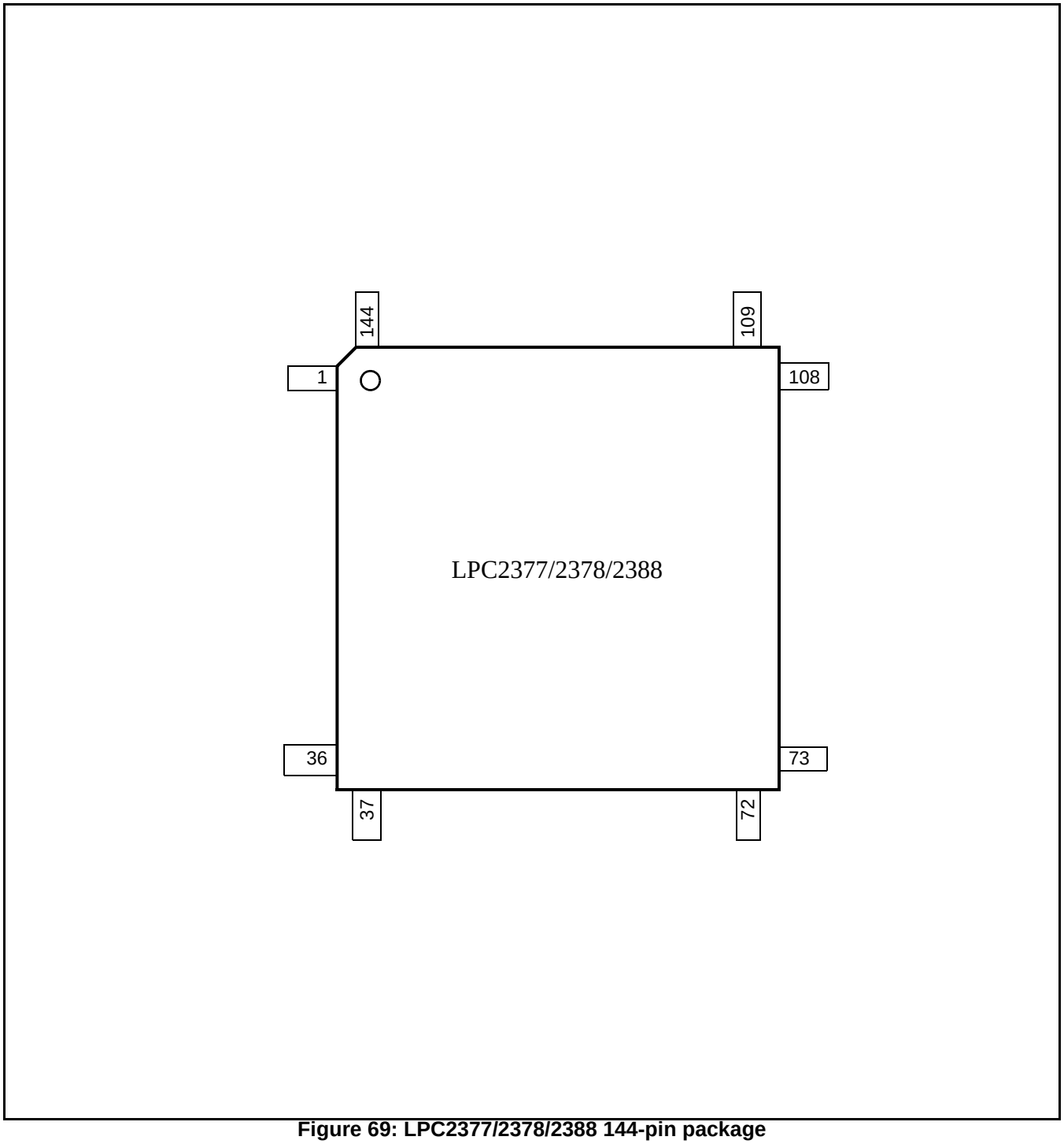
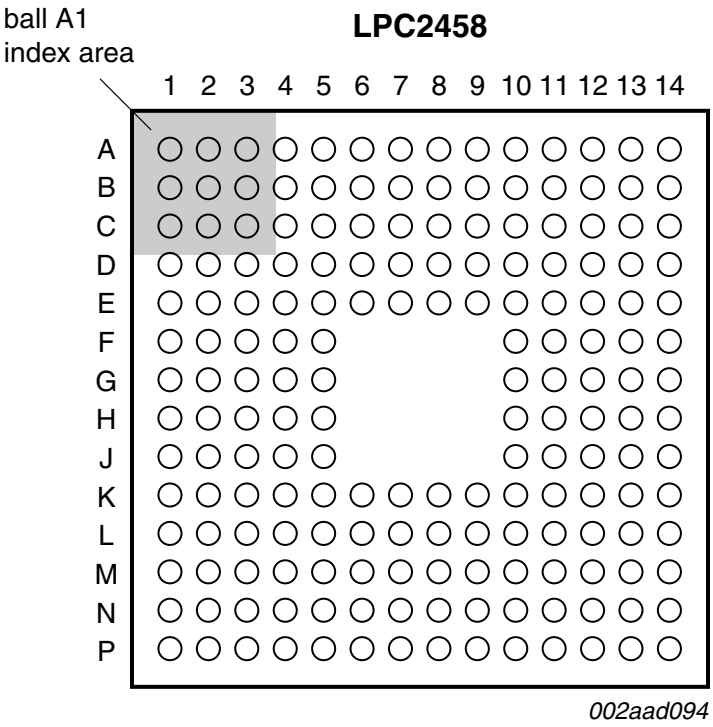


Figure 69: LPC2377/2378/2388 144-pin package

LPC2458 PINOUT



Transparent top view

Figure 70: LPC2458 180-pin TFBGA package

LPC2468/2478 PINOUT

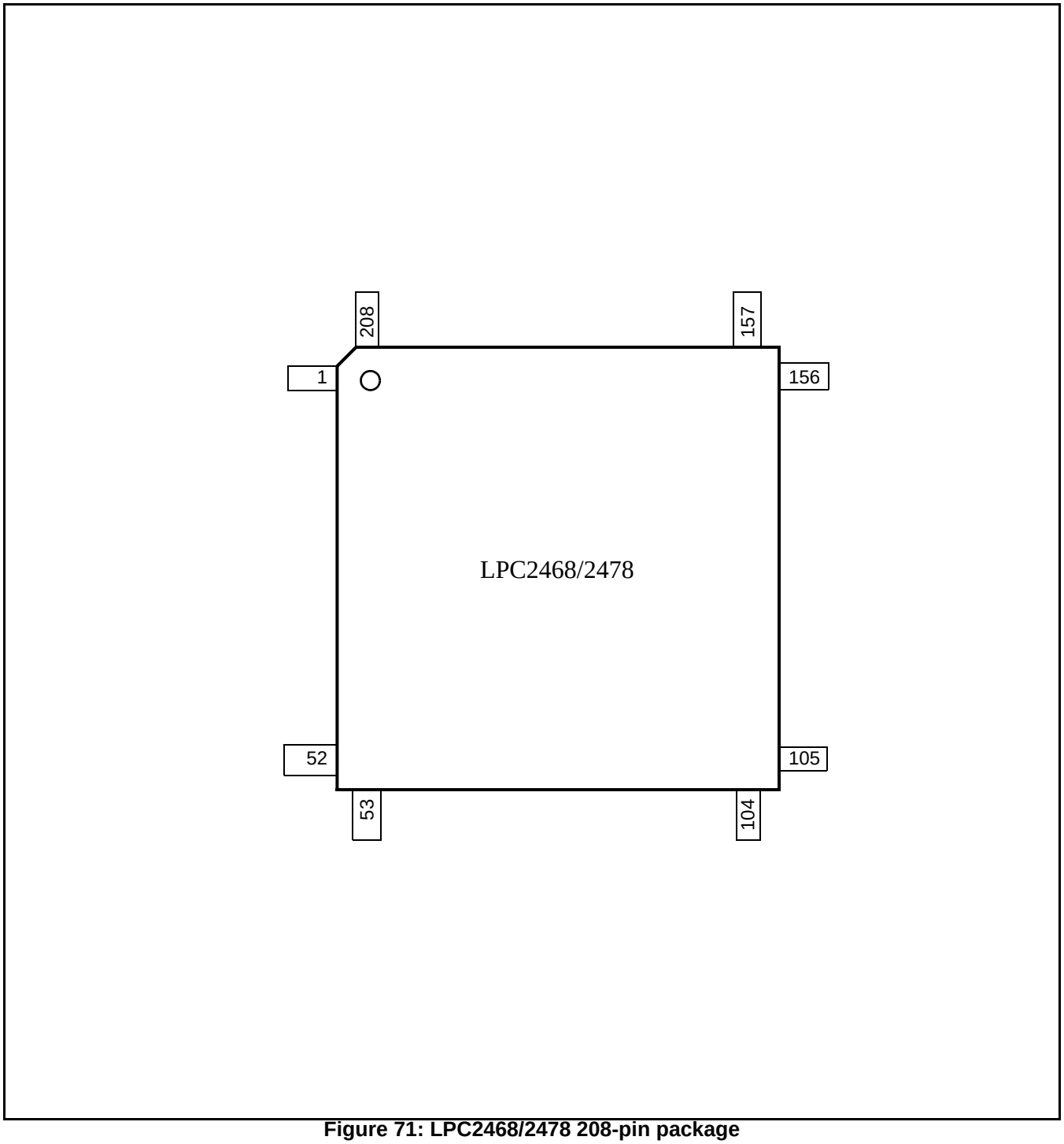
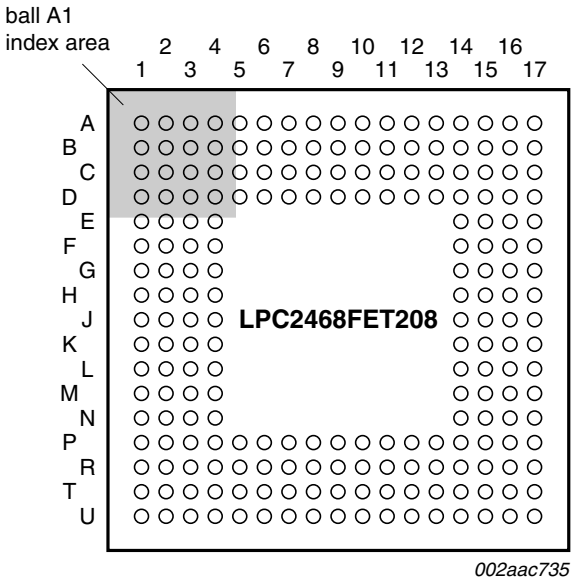
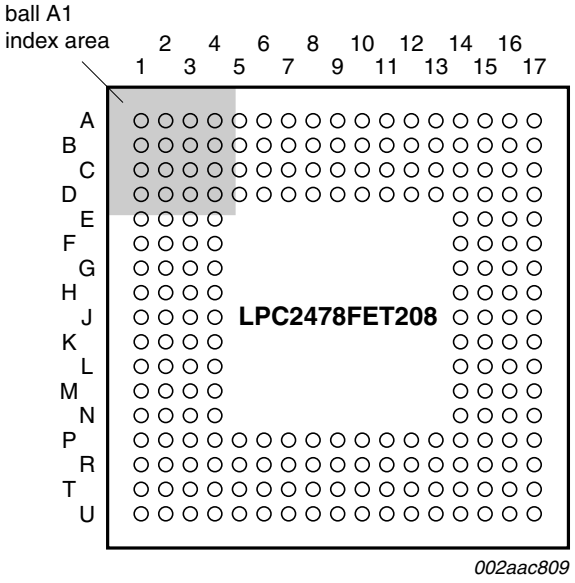


Figure 71: LPC2468/2478 208-pin package



Transparent top view



Transparent top view

Figure 72: LPC2468/2478 208-ball TFBGA package

LPC4072/4074/4076/4078/4088 PINOUT

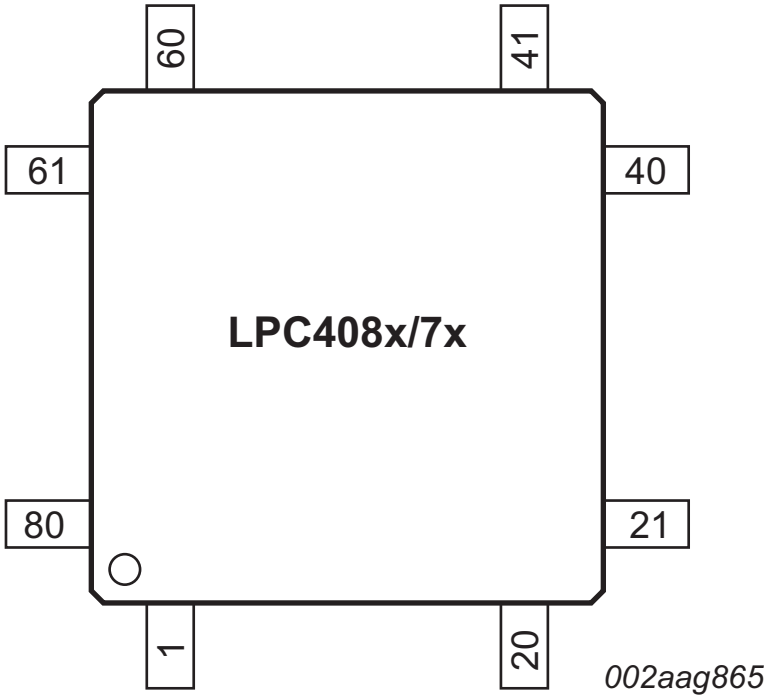


Figure 73: LPC4072/4074/4078 80-pin LQFP package

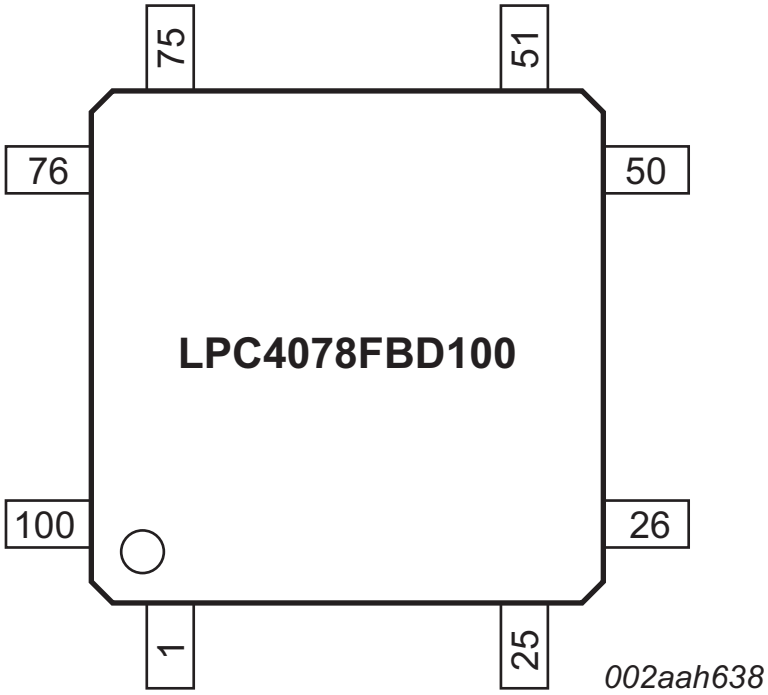


Figure 74: LPC4078 100-pin LQFP package

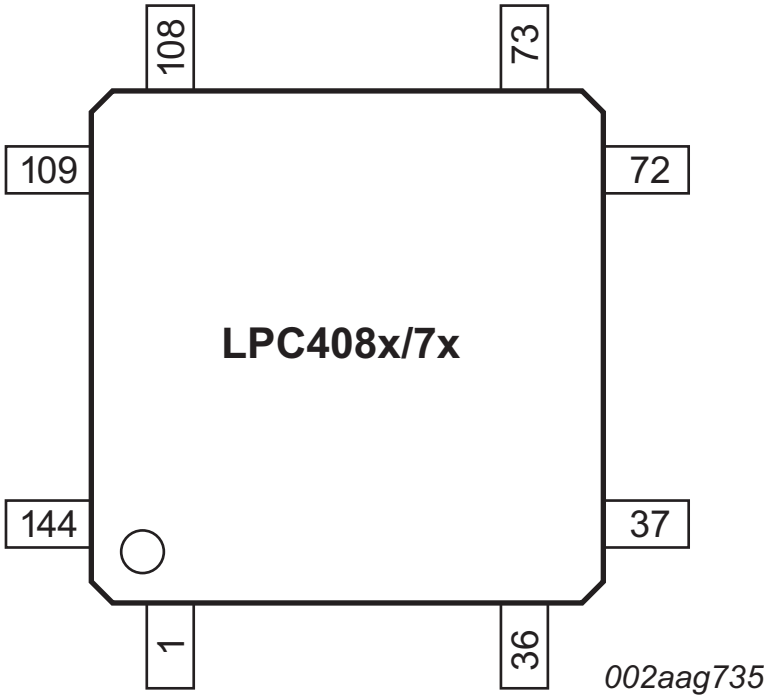


Figure 75: LP4074/4076/4078/4088 144-pin LQFP package

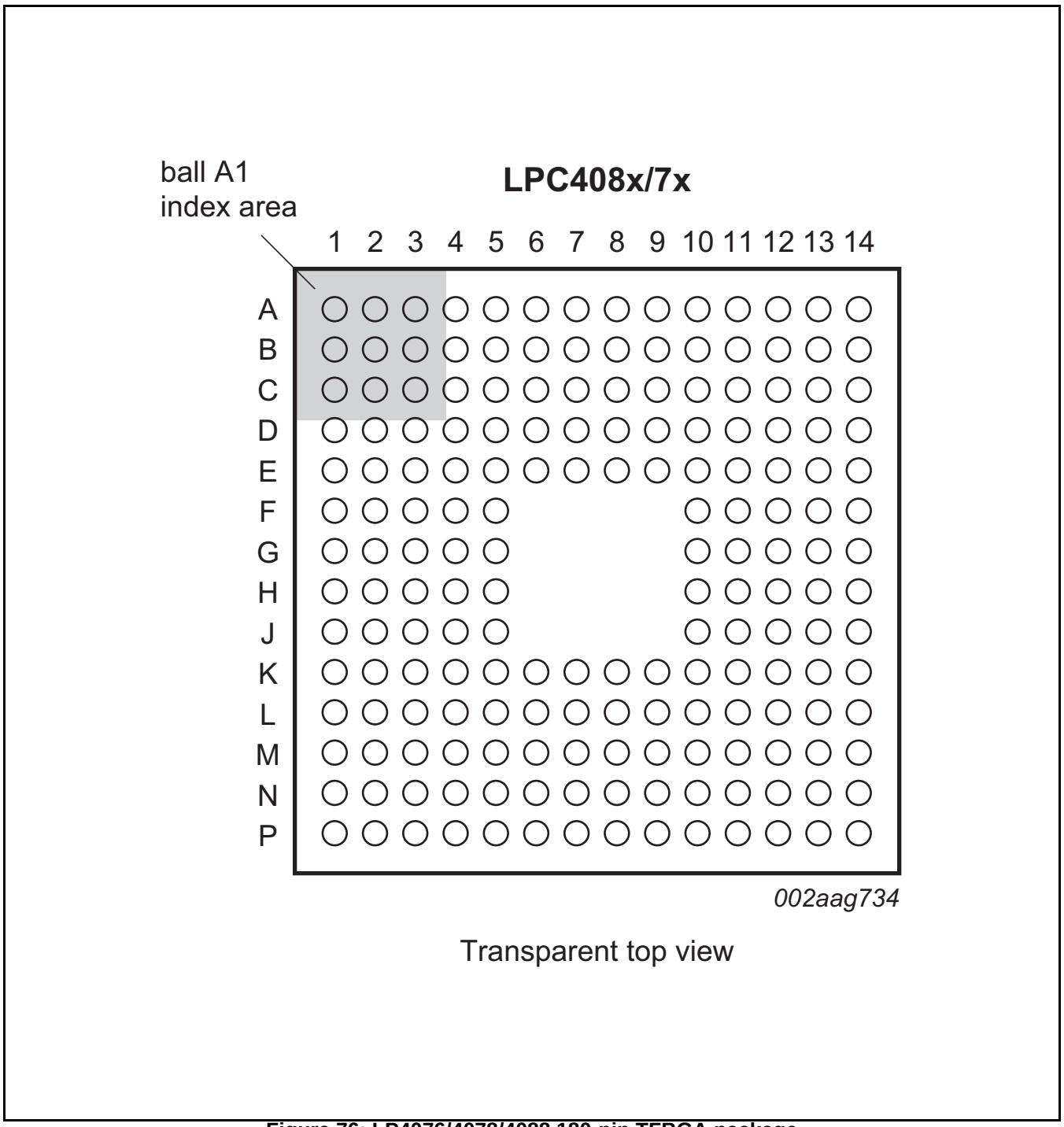


Figure 76: LP4076/4078/4088 180-pin TFBGA package

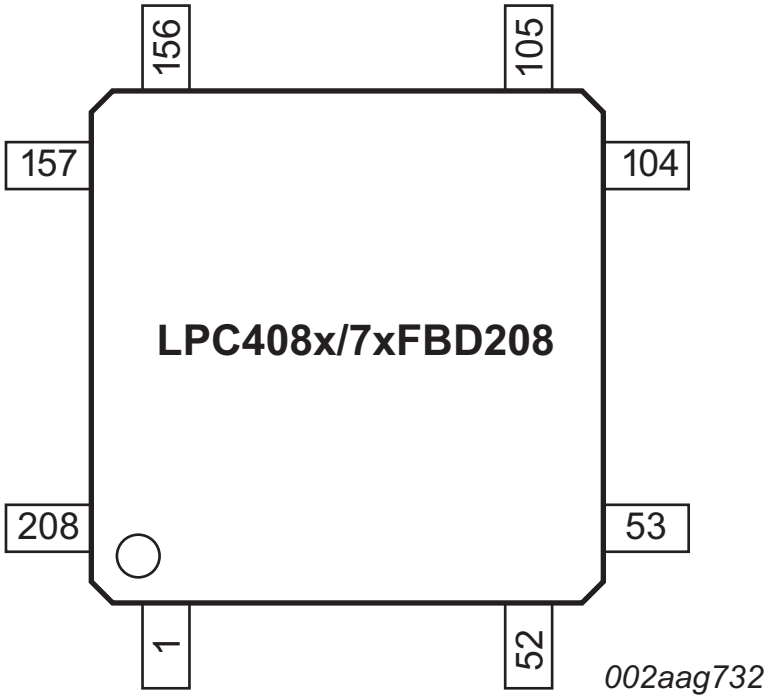


Figure 77: LP4078/4088 208-pin LQFP package

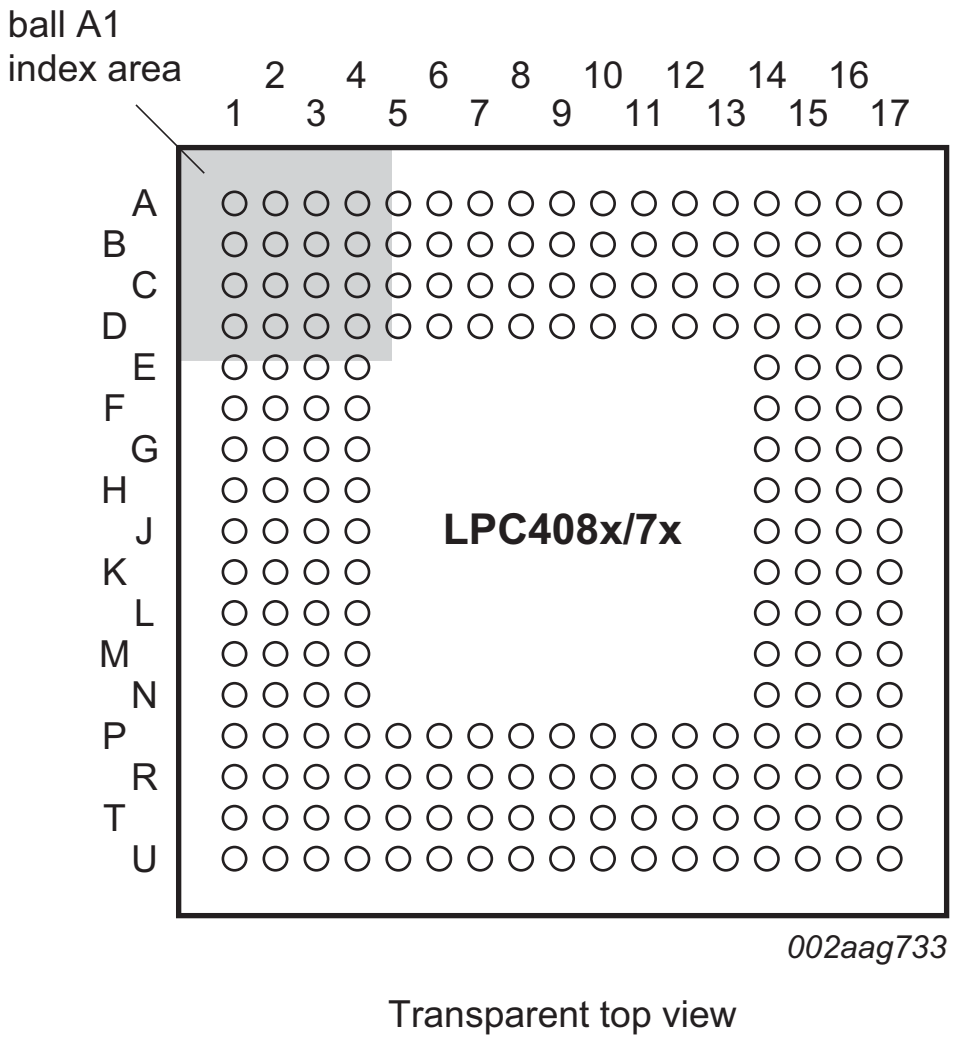


Figure 78: LP4078/4088 208-pin TFBGA package

LPC4312/4313/4315/4317/4322/4323/4325/4327/4333/4337/4353/4357 PINOUT

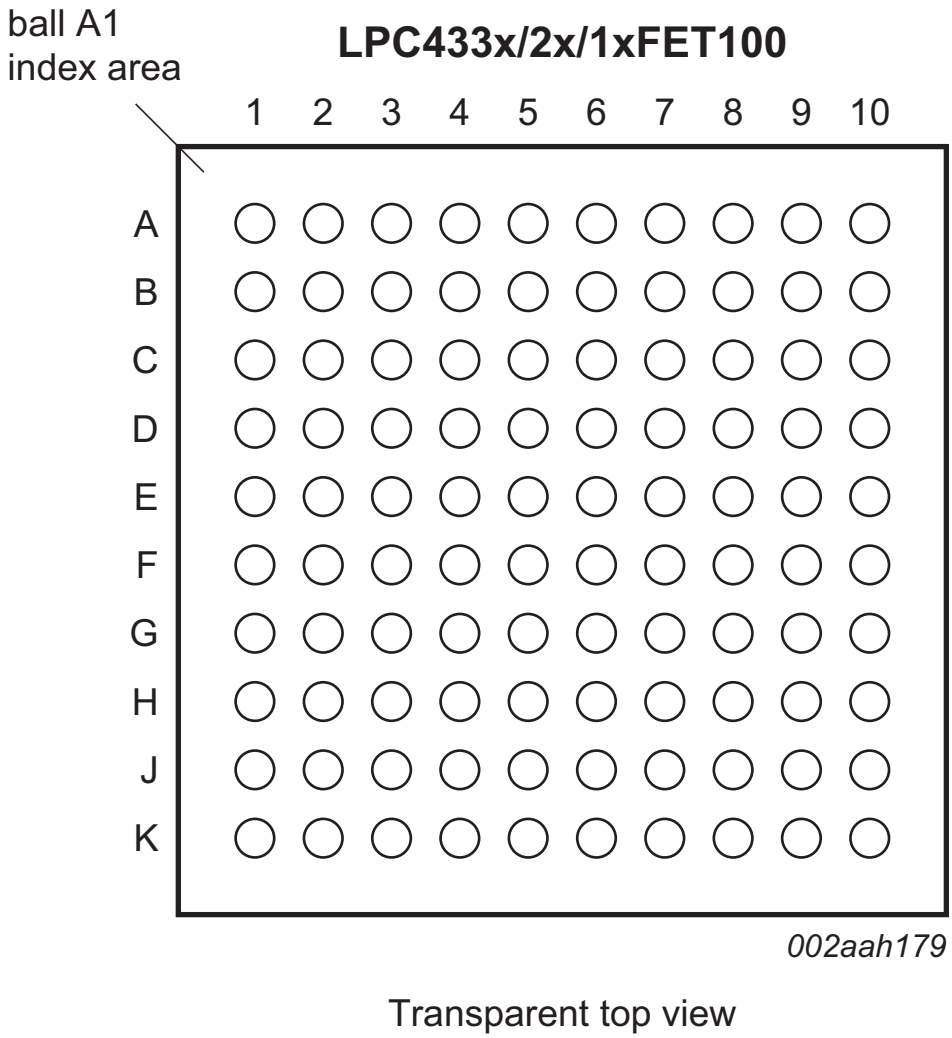


Figure 79: LPC4312/4313/4315/4317/4322/4323/4325/4327/4333/4337 100-pin TFBGA package

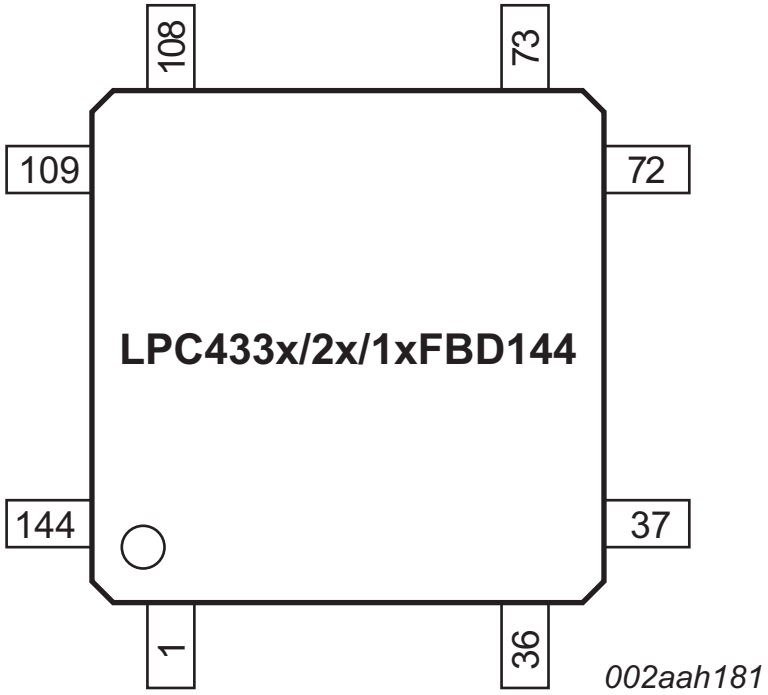


Figure 80: LPC4312/4313/4315/4317/4322/4323/4325/4327/4333/4337 144-pin LQFP package

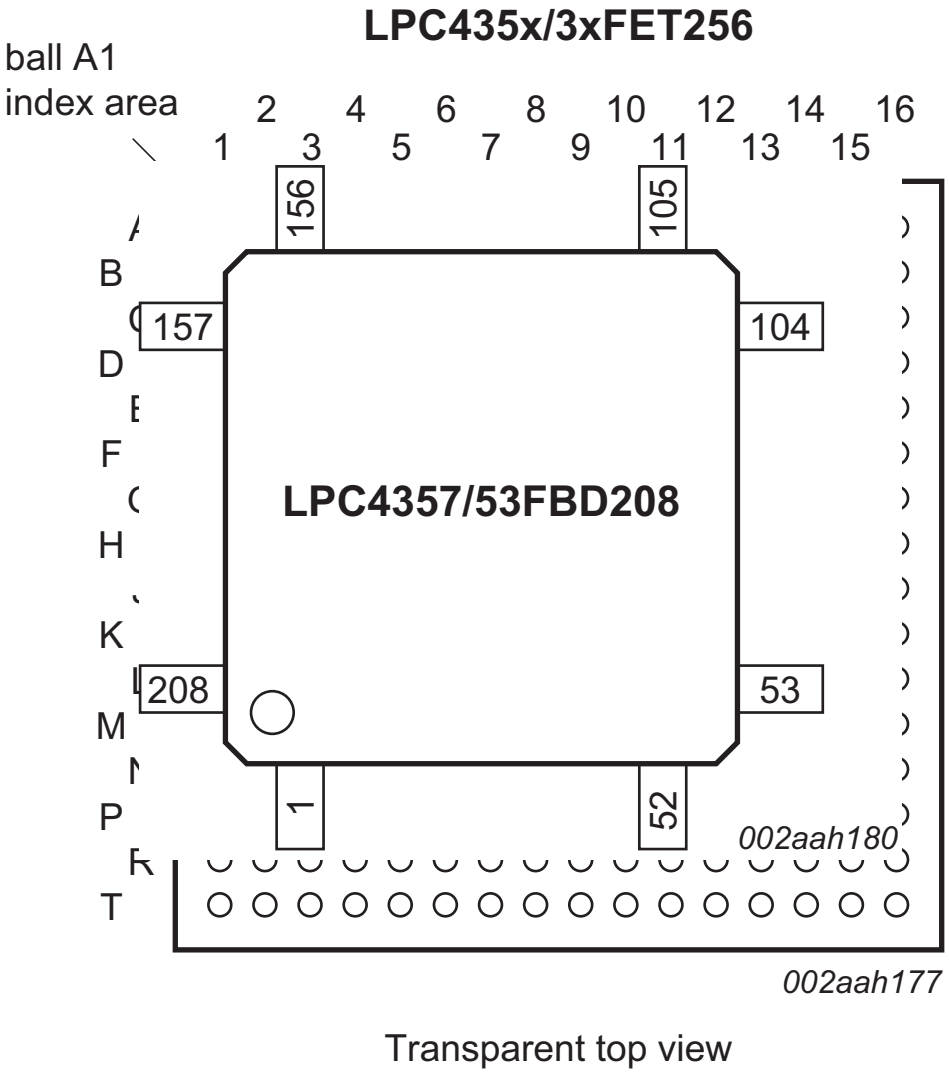


Figure 81: LP4353/4357 208-pin LQFP package

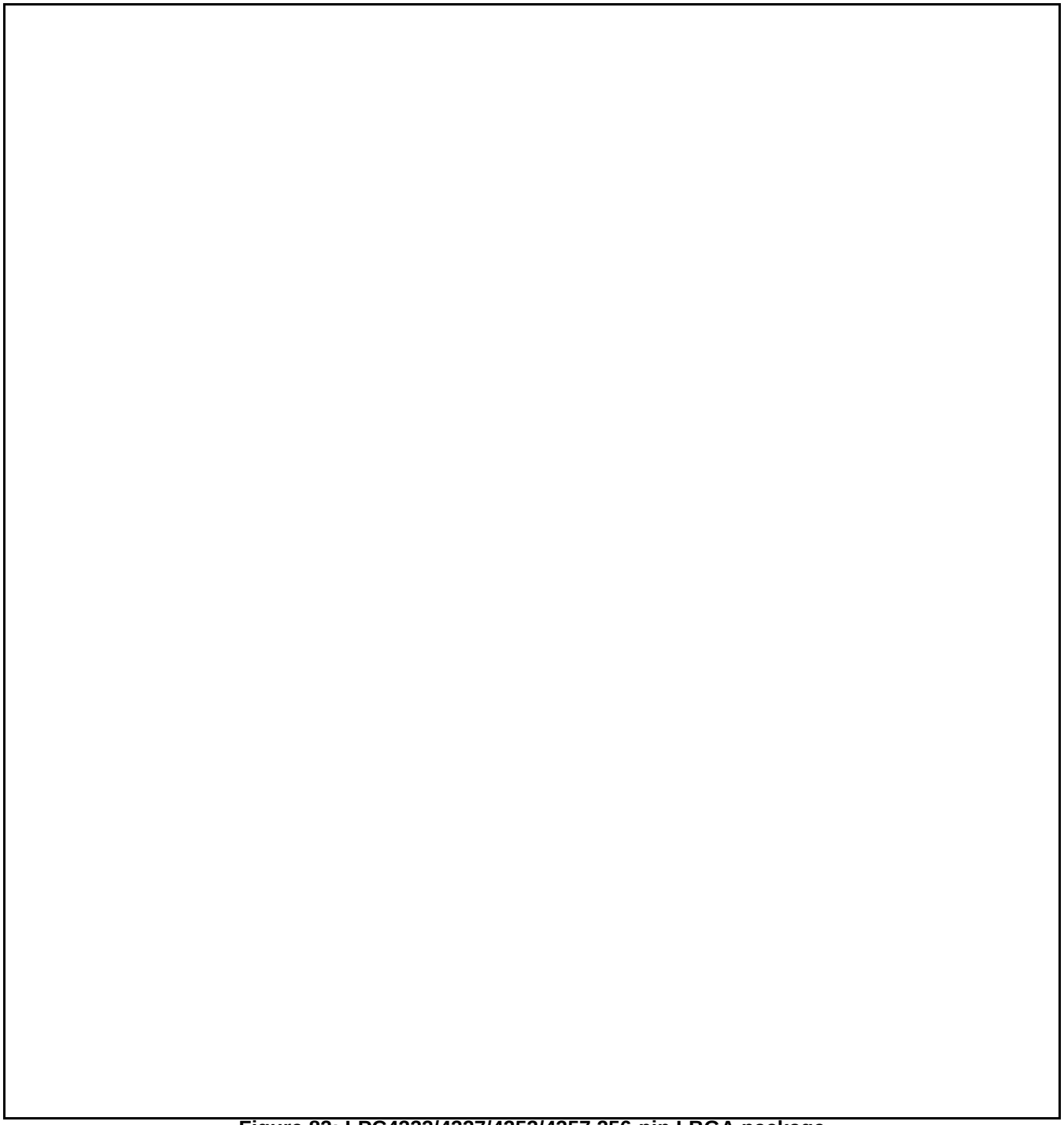


Figure 82: LPC4333/4337/4353/4357 256-pin LBGA package

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 2.0

Table 1: Pin names, description and functions - used with Boot Code Parallel 2.0

Pin Name	Name During Programming	I/O	Pin number				Function During Programming
			LPC2101/2/3, LPC2104/5/6 (48 pin package)	LPC2109/2114/19/24/29/ 2194, 2131/2/4/6/8, 2141/2/4/6/8 (64 pin package)	LPC2212/14/92/94 (144 pin package)	LPC2292 (144 ball TFBGA package)	
P0.0	SDOUT/ CMD	O/I	13	19	42	L4	Serial Data Out line and Command line
P0.1	SDIN/RES	I/O	14	21	49	K6	Serial Data In line and Response line
P0.2	HSC	O*	18	22	50	L6	Handshake line - Chip
P0.3	S0	O*	21	26	58	M8	Status line 0
P0.4	S1	O*	22	27	59	L8	Status line 1
P0.5	S2	O*	23	29	61	N9	Status line 2
P0.6	S3	O*	24	30	68	N11	Status line 3
P0.7	HSP	I	28	31	69	M11	Handshake line - Programmer
P0.8	D0	I/O	29	33	75	L12	Data line 0 (the LSB - the least significant bit in data)
P0.9	D1	I/O	30	34	76	L13	Data line 1
P0.10	D2	I/O	35	35	78	K11	Data line 2
P0.11[6]	D3	I/O	36	37	83	J12	Data line 3
P0.12	D4	I/O	37	38	84	J13	Data line 4
P0.13	D5	I/O	41	39	85	H10	Data line 5
P0.14[6]	D6/ISP	I/O	44	41	92	G10	Data line 6/ ISP activation line
P0.15	D7	I/O	45	45	99	E11	Data line 7 (the MSB - the most significant bit in data)
$\overline{RST}$	Reset	I	6	57	135	C5	Reset line used to reset the programmed part.
DBGSEL	DBGSEL	I	27	NA[3]	NA[3]	NA[3]	This line must be kept low all the time.
X1	X1	I	11	62	142	C3	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	0	12	61	141	B3	Output from the oscillator amplifier
V_{SSx}	VSS	I	7,19, 31,43	6,18,25,42, 50,59,58[4]	3,9,26,38, 54,67,79, 93,103,107, 111,128, 139,138	B4,B11,B13, C2,C4,D7, D11,E4,F13, J2,K12,L10, N2,N7	0 V reference[1]
$V_{DD1.8x}$	$V_{DD1.8x}$	I	5	17[5],49[5], 63[5]	37,110,143	A2,A12,N1	1.8 V Core/PLL Power Supply. Must be present during programming.

Table 1: Pin names, description and functions - used with Boot Code Parallel 2.0

Pin Name	Name During Programming	I/O	Pin number				Function During Programming
			LPC2101/2/3, LPC2104/5/6 (48 pin package)	LPC2109/2114/19/24/29/ 2194, 2131/2/4/6/8, 2141/2/4/6/8 (64 pin package)	LPC2212/14/92/94 (144 pin package)	LPC2292 (144 ball TFBGA package)	
V_{DD3x}	V_{DD3}	I	17,41	23,43, 51,7	2,14,31,39, 51,57,77, 94,104, 112,119	A11,B1,B9, C13,F3,F12, K3,K10,M3, M6,N8	3.3 V Pad power Supply[2]. This is the power supply voltage for the I/O ports.
Vbat[7]	4	I	4	-	-		For a LPC2101/2/3 the Vbat pin must be tied to either 1.8V or 3.3V during parallel programming. If the Vbat pin is grounded or left NC (not connected) the parallel programming will not work!

Note:
[*]pins P0.2 and P0.3 do not have an internal pull-up. For successful LPC2000 - parallel programmer communication, pins marked with [*] require an external pull-up of 6 kiloOhms.
[1]"low level" and "low" in the following text refers to 0V voltage level.
[2]"high level" and "high" in the following text refers to 3.3V voltage level.
[3]NA - not available in this microcontroller
[4]58 - pin 58 is not used as a V_{SS} by the LPC2131/2132/2134/2136/2138 and LPC2141/2142/2144/2146/2148 and can be left untied
[5]17,49,63 - these pins are not used as a power supply pins in LPC2131/2132/2134/2136/2138 microcontrollers and should not be tied; furthermore, pin 17 on LPC2131/2132/2134/2136/2138 and LPC2141/2142/2144/2146/2148 device MUST NOT be pulled low at any time, or the microcontroller will not operate properly
[6]For successful programming of the LPC2131/2132/2134/2136/2138 and LPC2141/2142/2144/2146/2148 devices pins P0.11 and P0.14 require an external pull-up of 6 kiloOhms
[7]This pin is used in LPC2101/2102/2103 parallel programming only.

The rest of the pins are not used in the programming and verification procedure and can be left untied.

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 3.0

Table 2: Pin names, description and functions - used with Boot Code Parallel 3.0 (LQFP packages)

Pin Name	Name During Programming	I/O	Pin number			Function During Programming
			LPC2361/62/64/6/65/66/67/68/87 (100 pin package)	LPC2377/78/88 (144 pin package)	LPC2468/78 (208 pin package)	
P0.0	CMD	I	46	66	94	Command line
P0.1	RES	O	47	67	96	Response line
P0.2	SDOUT	O	98	141	202	Serial Data Out line
P0.3	SDIN	I	99	142	204	Serial Data In line
P0.4	S0	O	81	116	168	Status line 0
P0.5	S1	O	80	115	166	Status line 1
P0.6	S2	O	79	113	164	Status line 2
P0.7	S3	O	78	112	162	Status line 3
P0.8	HSP	I	77	111	160	Handshake Line - Programmer
P0.9	HSC	O	76	109	158	Handshake Line - LPC2000 Chip
P2.10	ISP	I	53	76	110	ISP activation line
P2.0	D0	I/O	75	107	154	Data line 0 (the LSB - the least significant bit in data)
P2.1	D1	I/O	74	106	152	Data line 1
P2.2	D2	I/O	73	105	150	Data line 2
P2.3	D3	I/O	70	100	144	Data line 3
P2.4	D4	I/O	69	99	142	Data line 4
P2.5	D5	I/O	68	97	140	Data line 5
P2.6	D6	I/O	67	96	138	Data line 6
P2.7	D7	I/O	66	95	136	Data line 7
$\overline{RST}$	Reset	I	17	24	35	Reset line used to reset the programmed part.
DBGEN	DBGEN	I	NA	6	9	This line must be kept high all the time.
X1	X1	I	22	31	44	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	0	23	33	46	Output from the oscillator amplifier
V_{SSx}	VSS	I	11,15, 31,41, 55,72, 83,97	15,22,44,59, 65,79,103, 117,119,139	22,32,33,63,77, 84,93,114,133, 148,169,172, 189,200	0 V reference
V_{DD3x}	V_{DD3}	I	10,13,19, 28,42, 54,71, 84,96	14,18,27, 41,60,62, 77,102,114, 121,138	15,20,26,38, 60,71,86,89,112, 125,146,165,174, 181,198	3.3 V Power Supply
NC	NC	-	-	21,81,98	30,117,141	Leave these pins not connected/floating.

Table 3: Pin names, description and functions - used with Boot Code Parallel 3.0 (TFBGA packages)

Pin Name	Name During Programming	I/O	Ball number			Function During Programming
			LPC2364/68 (100 ball package)	LPC2458 (180 ball package)	LPC2468/78 (208 ball package)	
P0.0	CMD	I	K8	M10	U15	Command line
P0.1	RES	O	J8	N11	T14	Response line
P0.2	SDOUT	O	C4	D5	C4	Serial Data Out line
P0.3	SDIN	I	A2	A3	D6	Serial Data In line
P0.4	S0	O	A8	A11	B12	Status line 0
P0.5	S1	O	D7	B11	C12	Status line 1
P0.6	S2	O	B8	D11	D13	Status line 2
P0.7	S3	O	A9	B12	C13	Status line 3
P0.8	HSP	I	C8	C12	A15	Handshake Line - Programmer
P0.9	HSC	O	A10	A13	C14	Handshake Line - LPC2000 Chip
P2.10	ISP	I	J10	M13	N15	ISP activation line
P2.0	D0	I/O	B9	D12	B17	Data line 0 (the LSB - the least significant bit in data)
P2.1	D1	I/O	B10	C14	E14	Data line 1
P2.2	D2	I/O	D8	E11	D15	Data line 2
P2.3	D3	I/O	E7	E13	E16	Data line 3
P2.4	D4	I/O	D9	E14	D17	Data line 4
P2.5	D5	I/O	D10	F12	F16	Data line 5
P2.6	D6	I/O	E8	F13	E17	Data line 6
P2.7	D7	I/O	E9	G11	G16	Data line 7
$\overline{RST}$	Reset	I	F3	J1	M2	Reset line used to reset the programmed part.
DBGEN	DBGEN	I	D4	E5	F4	This line must be kept high all the time.
X1	X1	I	H2	L2	M4	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	0	G3	K4	N4	Output from the oscillator amplifier
V_{SSx}	VSS	I	B3, B7, C9, E1, F1, G7, J6, K3	A10, B4, C11, D13, F3,G13, H3, H4,L8,L9, L13,P4	A2,A12,B6, D12,E15,H14, J2,K4,L3, N16,P10,P12, R9,T5	0 V reference
V_{DD3x}	V_{DD3}	I	A3, A7, C10, E2, E4, G2, H6, H9, K2	C5, E2, E9, E10, E12, F2, G1, J14, K1, K8, L4, L11, N9	B13,C9,C17, D7,D11,G3, G4,H4,K16, M3,P6,P8, P11,P17,U13	3.3 V Power Supply
NC	NC	-	-	G10, H1, L12	G14, J4, L14	Leave these balls not connected/floating.

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 4.0

Table 4: Pin names, description and functions - used with Boot Code Parallel 4.0

Pin Name	Name During Programming	I/O	Pin number			Function During Programming
			LPC1751/2/4/6/8/9 (80 pin package)	LPC1763/4/5/6/7/8/9 (100 pin LQFP package)	LPC1765/8 (100 pin TFBGA package)	
P0.0	CMD	I	37	46	K8	Command line
P0.1	RES	O	38	47	J8	Response line
P0.2	SDOUT	O	79	98	C4	Serial Data Out line
P0.3	SDIN	I	80	99	A2	Serial Data In line
P0.6	HSP	I	64	79	B8	Handshake Line - Programmer
P0.7	HSC	O	63	78	A9	Handshake Line - LPC17xx Chip
P0.8	S0	O	62	77	C8	Status line 0
P0.9	S1	O	61	76	A10	Status line 1
P0.10	S2	O	39	48	H7	Status line 2
P0.11	S3	O	40	49	K9	Status line 3
P2.10	ISP	I	41	53	J10	ISP activation line
P2.0	D0	I/O	60	75	B9	Data line 0 (the LSB - the least significant bit in data)
P2.1	D1	I/O	59	74	B10	Data line 1
P2.2	D2	I/O	58	73	D8	Data line 2
P2.3	D3	I/O	55	70	E7	Data line 3
P2.4	D4	I/O	54	69	D9	Data line 4
P2.5	D5	I/O	53	68	D10	Data line 5
P2.6	D6	I/O	52	67	E8	Data line 6
P2.7	D7	I/O	51	66	E9	Data line 7
$\overline{RST}$	Reset	I	14	17	F3	Reset line used to reset the programmed part.
X1	X1	I	19	22	H2	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	0	20	23	G3	Output from the oscillator amplifier
V_{SSx}	VSS	I	9, 12, 24, 33, 43, 57, 66, 78	11, 15, 31, 41, 55, 72, 83, 97	B3, B7, C9, E1, F1, G7, J6, K3,	0 V reference
V_{DD3x}	V_{DD3}	I	8, 10, 16, 21, 42, 56, 77, 34, 67	10, 12, 19, 28, 54, 71, 96, 42, 84	A3, A7 C10, E2, E3, G2, H6, H9, K2	3.3 V Power Supply
NC	NC	-	-	13	D4, E4	Leave this pin not connected/floating.

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 5.0

Table 5: Pin names, description and functions - used with Boot Code Parallel 5.0

Pin Name	Name During Programming	I/O	Pin number		Function During Programming
			LPC1311/13/42/43 LPC1311/01 LPC1313/01 (33 pin package)	LPC1313/1343 LPC1313/01 (48 pin package)	
P0.6	CMD	I	15	22	Command line
P0.7	RES	O	16	23	Response line
P1.7	SDOUT	O	32	47	Serial Data Out line
P1.6	SDIN	I	31	46	Serial Data In line
P1.10	HSP	I	20	30	Handshake Line - Programmer
P1.11	HSC	O	27	42	Handshake Line - LPC1311/13/42/43xx Chip
P0.8	S0	O	17	27	Status line 0
P0.9	S1	O	18	28	Status line 1
P0.10	S2	O	19	29	Status line 2
P0.11	S3	O	21	32	Status line 3
P0.1	ISP	I	3	4	ISP activation line - all LPC1311/13/42/43 devices
P0.3	ISP	I	9	14	Additional ISP activation line - LPC1342/43 need two ISP lines!
P1.0	D0	I/O	22	33	Data line 0 (the LSB - the least significant bit in data)
P1.1	D1	I/O	23	34	Data line 1
P1.2	D2	I/O	24	35	Data line 2
P1.3	D3	I/O	25	39	Data line 3
P1.4	D4	I/O	26	40	Data line 4
P1.5	D5	I/O	30	45	Data line 5
P1.8	D6	I/O	7	9	Data line 6
P1.9	D7	I/O	12	17	Data line 7
$\overline{RST}$	Reset	I	2	3	Reset line used to reset the programmed part.
X1	X1	I	4	6	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	0	5	7	Output from the oscillator amplifier
V_{SS}	VSS	I	33	5, 41	0 V reference
$V_{DD(IO)}$/ (3V3)	V_{DD3}	I	6, 29	8, 44	3.3 V Power Supply

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 6.0

Table 6: Pin names, description and functions - used with Boot Code Parallel 6.0 (28/33 pin packages)

Pin Name	Name During Programming	I/O	Pin number		Function During Programming
			LPC1112/4 (28 pin package)	LPC1111/2/3/4 (33 pin package)	
P0.4	CMD	I	27	10	Command line
P1.5	RES	O	14	30	Response line
P1.7	SDOUT	O	16	32	Serial Data Out line
P1.6	SDIN	I	15	31	Serial Data In line
P0.5	HSP	I	5	11	Handshake Line - Programmer
P1.4	HSC	O	13	26	Handshake Line - LPC11xx Chip
P1.0	S0	O	9	22	Status line 0
P1.1	S1	O	10	23	Status line 1
P1.2	S2	O	11	24	Status line 2
P1.3	S3	O	12	25	Status line 3
P0.1	ISP	I	24	3	ISP activation line - all LPC11xx devices
P0.2	D0	I/O	25	8	Data line 0 (the LSB - the least significant bit in data)
P0.3	D1	I/O	26	9	Data line 1 **Important:** In case of LPC11C12/C14/C22/C24 microcontrollers this line must be held HIGH while the ISP activation line (P0.1) is LOW when the device is entering the Flash programming mode.
P0.6	D2	I/O	6	15	Data line 2
P0.7	D3	I/O	28	16	Data line 3
P0.8	D4	I/O	1	17	Data line 4
P0.9	D5	I/O	2	18	Data line 5
P0.10	D6	I/O	3	19	Data line 6
P0.11	D7	I/O	4	21	Data line 7
$\overline{RST}$	Reset	I	23	2	Reset line used to reset the programmed part.
X1	X1	I	20	4	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	0	19	5	Output from the oscillator amplifier
V_{SS}	VSS	I	8,22	33	0 V reference
$V_{DD(IO)}$/ (3V3)	V_{DD3}	I	7,21	6,29	3.3 V Power Supply

Table 7: Pin names, description and functions - used with Boot Code Parallel 6.0 (44/48/100 pin packages)

Pin Name	Name During Programming	I/O	Pin number				Function During Programming
			LPC11C12/C14 LPC1113/14/15 (48 pin package)	LPC11C22/C24 (48 pin package)	LPC1114 (44 pin package)	LPC11D14 (100 pin package)	
P0.4	CMD	I	15	15	19	18	Command line
P1.5	RES	O	45	45	2	99	Response line
P1.7	SDOUT	O	47	47	4	1	Serial Data Out line
P1.6	SDIN	I	46	46	3	100	Serial Data In line
P0.5	HSP	I	16	16	20	19	Handshake Line - Programmer
P1.4	HSC	O	40	40	42	94	Handshake Line - LPC11xx Chip
P1.0	S0	O	33	33	37	87	Status line 0
P1.1	S1	O	34	34	38	88	Status line 1
P1.2	S2	O	35	35	39	89	Status line 2
P1.3	S3	O	39	39	41	93	Status line 3
P0.1	ISP	I	4	4	8	7	ISP activation line - all LPC11xx devices
P0.2	D0	I/O	10	10	14	13	Data line 0 (the LSB - the least significant bit in data)
P0.3	D1	I/O	14	14	18	17	Data line 1 **Important:** In case of LPC11C12/C14/C22/C24 microcontrollers this line must be held HIGH while the ISP activation line (P0.1) is LOW when the device is entering the Flash programming mode.
P0.6	D2	I/O	22	23	26	25	Data line 2
P0.7	D3	I/O	23	24	27	26	Data line 3
P0.8	D4	I/O	27	27	31	81	Data line 4
P0.9	D5	I/O	28	28	32	82	Data line 5
P0.10	D6	I/O	29	29	33	83	Data line 6
P0.11	D7	I/O	32	32	36	86	Data line 7
$\overline{RST}$	Reset	I	3	3	7	6	Reset line used to reset the programmed part.
X1	X1	I	6	6	10	9	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	0	7	7	11	10	Output from the oscillator amplifier
V_{SS}	VSS	I	5,41	5, 41	9,43	8,40, 95	0 V reference
$V_{DD(IO)}$/ (3V3)	V_{DD3}	I	8,44	8, 44	1,12	11,39, 41,98	3.3 V Power Supply

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 6.1

Table 8: Pin names, description and functions - used with Boot Code Parallel 6.1

Pin Name	Name During Programming	I/O	Pin number				Function During Programming
			LPC11E11/E14 LPC11U12/U14/U24/U34/U35 LPC1315/16/17/45/46/47 (33 pin package)	LPC11E12/E13/E14 LPC11U12/U13/U14 LPC11U23/U24/ LPC11U35/U36/U37 LPC1315/16/17/45/46/47 (48 pin package)	LPC11U14/U24/U35 (48 ball package)	LPC11U24/E14/U35/U36/U37 LPC1317/47 (64 pin package)	
P0.4	CMD	I	10	15	G3	20	Command line
P0.20	RES	O	7	9	F2	11	Response line
P0.19	SDOUT	O	32	47	B2	62	Serial Data Out line
P0.18	SDIN	I	31	46	B3	61	Serial Data In line
P0.5	HSP	I	11	16	H3	21	Handshake Line - Programmer
P0.2	HSC	O	8	10	F1	13	Handshake Line - Microcontroller
P0.14	S0	O	24	35	B7	46	Status line 0
P0.15	S1	O	25	39	B6	52	Status line 1
P0.16	S2	O	26	40	A6	53	Status line 2
P0.17	S3	O	30	45	A3	60	Status line 3
P0.1	ISP	I	3	4	C2	5	ISP activation line - LPC11Exx/Uxx/1315...
P0.3	ISP	I	9	14	H2	19	Additional ISP activation line - LPC11Uxx and LPC1315/16/17/45/46/47 devices need two ISP lines! LPC11Exx devices do not use this line.
P0.6	D0	I/O	15	22	H6	29	Data line 0 (the LSB - the least significant bit in data)
P0.7	D1	I/O	16	23	G7	30	Data line 1
P0.8	D2	I/O	17	27	F8	36	Data line 2
P0.9	D3	I/O	18	28	F7	37	Data line 3
P0.10	D4	I/O	19	29	E7	38	Data line 4
P0.11	D5	I/O	21	32	D8	42	Data line 5
P0.12	D6	I/O	22	33	C7	44	Data line 6
P0.13	D7	I/O	23	34	C8	45	Data line 7
$\overline{\text{RESET}}$ P0.0	Reset	I	2	3	C1	4	Reset line used to reset the programmed part.
X1	X1	I	4	6	D1	8	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	0	5	7	E1	9	Output from the oscillator amplifier
V_{SS}	VSS	I	33	5, 41	B5, D2	7, 54	0 V reference
$V_{DD(IO)}$/ (3V3)	V_{DD3}	I	6, 29	8, 44	B4, E2	10, 33, 48,58	3.3 V Power Supply

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 6.2

Table 9: Pin names, description and functions - used with Boot Code Parallel 6.2

Pin Name	Name During Programming	I/O	Pin number			Function During Programming
			LPC1A02/A04 (20 pin package)	LPC11A11/A12/A13/A14 (33 pin package)	LPC11A12/A14 (48 pin package)	
P0.2	CMD	I	A1	10	15	Command line
P0.14	RES	O	B4	20	30	Response line
P0.13	SDOUT	O	D1	32	47	Serial Data Out line
P0.12	SDIN	I	E1	31	46	Serial Data In line
P0.3	HSP	I	B1	11	16	Handshake Line - Programmer
P0.15	HSC	O	E4	27	41	Handshake Line - LPC11Axx Chip
P0.18	S0	O	-[1]	8	10	Status line 0
P0.19	S1	O	-[1]	9	14	Status line 1
P0.20	S2	O	-[1]	15	22	Status line 2
P0.21	S3	O	-[1]	16	23	Status line 3
P0.1	ISP	I	B2	3	4	ISP activation line - all LPC11Axx devices
P0.4	D0	I/O	A4	18	28	Data line 0 (the LSB - the least significant bit in data)
P0.5	D1	I/O	B3	19	29	Data line 1
P0.6	D2	I/O	C3	21	32	Data line 2
P0.7	D3	I/O	C4	22	33	Data line 3
P0.8	D4	I/O	C2	23	34	Data line 4
P0.9	D5	I/O	D4	24	35	Data line 5
P0.10	D6	I/O	D3	25	38	Data line 6
P0.11	D7	I/O	D2	26	39	Data line 7
$\overline{\text{RESET}}$ P0.0	Reset	I	C1	2	3	Reset line used to reset the programmed part.
-	X1	I	-[2]	-[2]	-[2]	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
-	X2	O	-[2]	-[2]	-[2]	Output from the oscillator amplifier
V_{SS}	VSS	I	E3	33	5, 42	0 V reference
$V_{DD(IO)}$/ (3V3)	V_{DD3}	I	E2	6, 29	8, 44	3.3 V Power Supply

Note:
[1] 20 pin LPC11Axx devices do not provide Status lines
[2] LPC11Axx devices use the internal 12 MHz IRC oscillator for Flash programming and do not need an external clock

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 6.3

Table 10: Pin names, description and functions - used with Boot Code Parallel 6.3 (LPC1111/12)

Pin Name	Name During Programming	I/O	Pin number	Function During Programming
			LPC1111/1112 (20 pin package)	
P0.1	CMD	I	18	Command line (shared line with ISP)
P0.2	RES	O	19	Response line
P1.7	SDOUT	O	12	Serial Data Out line (shared line with HSC)
P1.6	SDIN	I	11	Serial Data In line (shared line with HSP)
P1.6	HSP	I	11	Handshake Line - Programmer (shared line with SDIN)
P1.7	HSC	O	12	Handshake Line - LPC1111/12 Chip (shared line with SDOUT)
-[1]	S0	O	-	Status line 0 - not available on the 20 pin LPC1111/12 devices
-[1]	S1	O	-	Status line 1 - not available on the 20 pin LPC1111/12 devices
-[1]	S2	O	-	Status line 2 - not available on the 20 pin LPC1111/12 devices
-[1]	S3	O	-	Status line 3 - not available on the 20 pin LPC1111/12 devices
P0.1	ISP	I	18	ISP activation line (shared line with CMD)
P1.0	D0	I/O	7	Data line 0 (the LSB - the least significant bit in data)
P1.1	D1	I/O	8	Data line 1
P1.2	D2	I/O	9	Data line 2
P1.3	D3	I/O	10	Data line 3
P0.8	D4	I/O	1	Data line 4
P0.9	D5	I/O	2	Data line 5
P0.10	D6	I/O	3	Data line 6
P0.11	D7	I/O	4	Data line 7
$\overline{\text{RESET}}$ P0.0	Reset	I	17	Reset line used to reset the programmed part.
X1	X1	I	14	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	O	13	Output from the oscillator amplifier
V_{SS}	VSS	I	6[2], 16	0 V reference
$V_{DD(IO)}$/ (3V3)	V_{DD3}	I	5[2], 15	3.3 V Power Supply

Note:
[1] 20 pin LPC1111/1112 devices do not provide Status lines
[2] LPC1112FDH20/102 only

Table 11: Pin names, description and functions - used with Boot Code Parallel 6.3 (LPC1101LV/02LV/12LV/14LV)

Pin Name	Name During Programming	I/O	Pin number			Function During Programming
			LPC1101LV/1102LV (25 pin package)	LPC1112LV/1114LV (24 pin package)	LPC1114LV (33 pin package)	
P0.1	CMD	I	C3	3	3	Command line (shared line with ISP)
P0.2	RES	O	B2	7	8	Response line
P1.7	SDOUT	O	E1	1	32	Serial Data Out line (shared line with HSC)
P1.6	SDIN	I	D2	24	31	Serial Data In line (shared line with HSP)
P1.6	HSP	I	D2	24	31	Handshake Line - Programmer (shared line with SDIN)
P1.7	HSC	O	E1	1	32	Handshake Line - LPC11xxLV Chip (shared line with SDOUT)
-[1]	S0	O	NA	NA	NA	Status line 0 - not available on the LPC11xxLV devices
-[1]	S1	O	NA	NA	NA	Status line 1 - not available on the LPC11xxLV devices
-[1]	S2	O	NA	NA	NA	Status line 2 - not available on the LPC11xxLV devices
-[1]	S3	O	NA	NA	NA	Status line 3 - not available on the LPC11xxLV devices
P0.1	ISP	I	C3	3	3	ISP activation line (shared line with CMD)
P1.0	D0	I/O	C4	16	22	Data line 0 (the LSB - the least significant bit in data)
P1.1	D1	I/O	D5	17	23	Data line 1
P1.2	D2	I/O	D4	18	24	Data line 2
P1.3	D3	I/O	E5	19	25	Data line 3
P0.8	D4	I/O	A5	12	17	Data line 4
P0.9	D5	I/O	B5	13	18	Data line 5
P0.10	D6	I/O	B4	14	19	Data line 6
P0.11	D7	I/O	C5	15	21	Data line 7
$\overline{\text{RESET}}$ P0.0	Reset	I	D1	2	2	Reset line used to reset the programmed part.
X1	X1	I	C1	4	4	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	O	C2	5	5	Output from the oscillator amplifier
V_{SS}	VSS	I	E4	21	33	0 V reference
$V_{DD(IO)}$	V_{DD3}	I	E3	22	6, 28, 29	1.8 V Power Supply

Note:
[1] LPC1101/LV/02LV/12LV/14LV devices do not provide Status lines

IMPORTANT: LPC1101LV/02LV/12LV/14LV ARE 1.8V ONLY MICROCONTROLLERS! NEITHER THE POWER SUPPLY NOR ANY OTHER SIGNAL APPLIED TO THESE DEVICES SHOULD EVER EXCEED THIS LIMIT!

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 7.0

Table 12: Pin names, description and functions - used with Boot Code Parallel 7.0

Pin Name	Name During Programming	I/O	Pin number	Function During Programming
			LPC1112FHN24 (24 pin package)	
P0.6	CMD	I	10	Command line
P1.4	RES	O	20	Response line
P1.7	SDOUT	O	24	Serial Data Out line
P1.6	SDIN	I	23	Serial Data In line
P0.7	HSP	I	11	Handshake Line - Programmer
P1.8	HSC	O	6	Handshake Line - LPC11xx Chip
-[1]	S0	O	-	Status line 0 - not available on the 24 pin LPC1112 device
-[1]	S1	O	-	Status line 1 - not available on the 24 pin LPC1112 device
-[1]	S2	O	-	Status line 2 - not available on the 24 pin LPC1112 device
-[1]	S3	O	-	Status line 3 - not available on the 24 pin LPC1112 device
P0.1	ISP	I	2	ISP activation line
P1.0	D0	I/O	16	Data line 0 (the LSB - the least significant bit in data)
P1.1	D1	I/O	17	Data line 1
P1.2	D2	I/O	18	Data line 2
P1.3	D3	I/O	19	Data line 3
P0.8	D4	I/O	12	Data line 4
P0.9	D5	I/O	13	Data line 5
P0.10	D6	I/O	14	Data line 6
P0.11	D7	I/O	15	Data line 7
$\overline{RST}$	Reset	I	1	Reset line used to reset the programmed part.
-[2]	X1	I	-	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
-[2]	X2	O	-	Output from the oscillator amplifier
V_{SS}	VSS	I	3, 21	0 V reference
$V_{DD(IO)}$/ (3V3)	V_{DD3}	I	5, 22	3.3 V Power Supply

Note:
[1] LPC1112FHN24 devices do not provide Status lines
[2] LPC1112FHN24 devices use the internal 12 MHz IRC oscillator for Flash programming and do not need an external clock

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 8.0

Table 13: Pin names, description and functions - used with Boot Code Parallel 8.0

Pin Name	Name During Programming	I/O	Pin number			Function During Programming
			LPC1224/5/6/7 (48 pin package)	LPC1224/5/6/7 (64 pin package)	LPC12D27 (100 pin package)	
P0.0	CMD	I	15	19	6	Command line
P0.4	RES	O	19	23	10	Response line
P0.2	SDOUT	O	17	21	8	Serial Data Out line
P0.1	SDIN	I	16	20	7	Serial Data In line
P0.3	HSP	I	18	22	9	Handshake Line - Programmer
P0.5	HSC	O	20	24	11	Handshake Line - LPC12xx Chip
P0.6	S0	O	21	25	12	Status line 0
P0.7	S1	O	22	26	13	Status line 1
P0.8	S2	O	23	27	14	Status line 2
P0.9	S3	O	24	28	15	Status line 3
P0.12	ISP	I	27	39	19	ISP activation line - all LPC12xx devices
P0.14	D0	I/O	29	41	21	Data line 0 (the LSB - the least significant bit in data)
P0.15	D1	I/O	30	42	22	Data line 1
P0.16	D2	I/O	31	43	23	Data line 2
P0.17	D3	I/O	32	44	24	Data line 3
P0.18	D4	I/O	33	45	25	Data line 4
P0.19	D5	I/O	4	4	95	Data line 5
P0.20	D6	I/O	5	5	96	Data line 6
P0.21	D7	I/O	6	6	97	Data line 7
$\overline{RST}$	Reset	I	28	40	20	Reset line used to reset the programmed part.
X1	X1	I	1	1	92	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	0	2	2	93	Output from the oscillator amplifier
V_{SS}	VSS	I	43,48	55,64	40,86, 91	0 V reference
$V_{DD(IO)}$/ (3V3)	V_{DD3}	I	44,47	56,63	39,41, 87,90	3.3 V Power Supply

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 9.0

Table 14: Pin names, description and functions - used with Boot Code Parallel 9.0 (208/180 pin packages)

Pin Name	Name During Programming	I/O	Pin number			Function During Programming
			LQFP 208 pin package	TFBGA 208 ball package	TFBGA 180 ball package	
P0.0	CMD	I	94	U15	M10	Command line
P0.1	RES	O	96	T14	N11	Response line
P0.2	SDOUT	O	202	C4	D5	Serial Data Out line
P0.3	SDIN	I	204	D6	A3	Serial Data In line
P0.6	HSP	I	164	D13	D11	Handshake Line - Programmer
P0.7	HSC	O	162	C13	B12	Handshake Line - LPC177x/8x and LPC407x/8x Chip
P0.8	S0	O	160	A15	C12	Status line 0
P0.9	S1	O	158	C14	A13	Status line 1
P0.10	S2	O	98	T15	L10	Status line 2
P0.11	S3	O	100	R14	P12	Status line 3
P2.10	ISP	I	110	N15	M13	ISP activation line - all LPC177x/8x and LPC407x/8x devices
P2.0	D0	I/O	154	B17	D12	Data line 0 (the LSB - the least significant bit in data)
P2.1	D1	I/O	152	E14	C14	Data line 1
P2.2	D2	I/O	150	D15	E11	Data line 2
P2.3	D3	I/O	144	E16	E13	Data line 3
P2.4	D4	I/O	142	D17	E14	Data line 4
P2.5	D5	I/O	140	F16	F12	Data line 5
P2.6	D6	I/O	138	E17	F13	Data line 6
P2.7	D7	I/O	136	G16	G11	Data line 7
$\overline{RST}$	Reset	I	35	M2	J1	Reset line used to reset the programmed part.
X1	X1	I	44	M4	L2	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	0	46	N4	K4	Output from the oscillator amplifier
V_{SS}	VSS	I	22,32,33,63, 77,84,93,114, 133,148,169, 172,189,200	A2,A12,B6, D12,E15,H14, J2,K4,L3,N16, P10,P12,R9,T5	A10,B4,C11, D13,F3,G13, H3,H4,L8,L9, L13,P4	0 V reference
$V_{DD(IO)}$ /(3V3)	V_{DD3}	I	15,20,24,26, 38,60,71,86, 89,112,125, 146,165,174, 181,198	B13,C9,C17, D7,D11,G3, G4,H4,K1, K16,M3,P6, P8,P11, P17,U13	C5,E2,E9, E10,E12,F2, G1,G2,J14, K1,K8,L4, L11,N9	3.3 V Power Supply

Table 15: Pin names, description and functions - used with Boot Code Parallel 9.0 (144/100/80 pin packages)

Pin Name	Name During Programming	I/O	Pin number			Function During Programming
			LQFP 144 pin package	LQFP 100 pin package	LQFP 80 pin package	
P0.0	CMD	I	66	46	37	Command line
P0.1	RES	O	67	47	38	Response line
P0.2	SDOUT	O	141	98	79	Serial Data Out line
P0.3	SDIN	I	142	99	80	Serial Data In line
P0.6	HSP	I	113	79	64	Handshake Line - Programmer
P0.7	HSC	O	112	78	63	Handshake Line - LPC177x/8x and LPC407x/8x Chip
P0.8	S0	O	111	77	62	Status line 0
P0.9	S1	O	109	76	61	Status line 1
P0.10	S2	O	69	48	39	Status line 2
P0.11	S3	O	70	49	40	Status line 3
P2.10	ISP	I	76	53	41	ISP activation line - all LPC177x/8x and LPC407x/8x devices
P2.0	D0	I/O	107	75	60	Data line 0 (the LSB - the least significant bit in data)
P2.1	D1	I/O	106	74	59	Data line 1
P2.2	D2	I/O	105	73	58	Data line 2
P2.3	D3	I/O	100	70	55	Data line 3
P2.4	D4	I/O	99	69	54	Data line 4
P2.5	D5	I/O	97	68	53	Data line 5
P2.6	D6	I/O	96	67	52	Data line 6
P2.7	D7	I/O	95	66	51	Data line 7
$\overline{RST}$	Reset	I	24	17	14	Reset line used to reset the programmed part.
X1	X1	I	31	22	19	Input to the oscillator circuit and internal clock generator circuits. Must be 10 MHz.
X2	X2	0	33	23	20	Output from the oscillator amplifier
V_{SS}	VSS	I	15,22,44,59, 65,79,103, 117,119,139	11, 15, 31, 41, 55, 72, 83, 97	9, 24, 33, 43, 57, 66, 78	0 V reference
$V_{DD(IO)}$ /(3V3)	V_{DD3}	I	14,17,18,27, 41,60,62, 77,102,114, 121,138	10, 12, 13, 19, 28, 42, 54, 71, 84, 96	8, 10, 16, 21, 34, 42, 56, 67, 77	3.3 V Power Supply

PINS USED IN PARALLEL PROGRAMMING WITH BOOT CODE (PARALLEL) 10.0

Table 16: Pin names, description and functions - used with Boot Code Parallel 10.0

Pin Name	Name During Programming	I/O	Package pin/ball number				Function During Programming
			TFBGA 100 ball	LQFP 144 pin	LQFP 208 pin	LBGA 256 ball	
P2.2	CMD	I	F5	84	121	M15	Command line
P6.1	RES	O	G5	74	107	R15	Response line
P2.0	SDOUT	O	G10	75	108	T16	Serial Data Out line
P2.1	SDIN	I	G7	81	116	N15	Serial Data In line
P2.4	HSP	I	D9	88	128	K11	Handshake Line - Programmer
P6.4	HSC	O	F6	80	114	R16	Handshake Line - LPC18xx/43xx Chip
P1.1	BOOT0	I	K2	42	58	R2	must be set to 0V by the programmer at reset and during programming
P1.2	BOOT1	I	K1	43	60	R3	must be set to 0V by the programmer at reset and during programming
P2.9	BOOT3	I	B10	102	144	H16	must be set to 0V by the programmer at reset and during programming
P2.5	S0	O	D10	91	131	K14	Status line 0
P2.6	S1	O	G9	95	137	K16	Status line 1
P2.8	BOOT2/ S2	I/O	C6	98	140	J16	must be set to 0V by the programmer at reset and kept at 0V until “?“ is received; following this it becomes Status line 2
P3.1	S3	O	F7	114	163	G11	Status line 3
P2.7	ISP	I	C10	96	138	H14	ISP activation line - all LPC18xx/43xx devices
P1.7	D0	I/O	G4	50	69	T5	Data line 0 (the LSB - the least significant bit in data)
P1.8	D1	I/O	H5	51	71	R7	Data line 1
P1.9	D2	I/O	J5	52	73	T7	Data line 2
P1.10	D3	I/O	H6	53	75	R8	Data line 3
P1.11	D4	I/O	J7	55	77	T9	Data line 4
P1.12	D5	I/O	K7	56	78	R9	Data line 5
P1.13	D6	I/O	H8	60	83	R10	Data line 6
P1.14	D7	I/O	J8	61	85	R11	Data line 7
$\overline{RST}$	Reset	I	B6	128	185	D9	Reset line used to reset the programmed part.
$V_{SS(IO)}$	VSS	I	C2,C8,D4, D5,G8,J3, J6	4,40,76, 109,135	5,56,109, 157,196	G9,H7,J10, J11,K8,C4, D13,G6,G7, G8,H8,H9, J8,J9,K9,K10, M13,P7,P13,B2	0 V reference

Table 16: Pin names, description and functions - used with Boot Code Parallel 10.0

Pin Name	Name During Programming	I/O	Package pin/ball number				Function During Programming
			TFBGA 100 ball	LQFP 144 pin	LQFP 208 pin	LBGA 256 ball	
$V_{DD(IO)}$/ (REG)/	V_{DD3}	I	B2,F10,K5, E4,E5,F4	137,5,36, 41,71,77, 107,111,141, 94,131,59, 25	198,6,52, 57,102,110, 155,160,202, 135,188,195, 82,33	B4,D7,E12, F7,F8,G10, H10,J6,J7, K7,L9,L10, N7,N13,F10, F9,L8,L7	3.3 V Power Supply
Important: LPC181x/2x/3x/5x and LPC431x/2x/3x/5x devices use an internal 12 MHz oscillator to generate a system clock of 96 MHz during parallel programming. Therefore there is no need for external crystal to be used when programming these microcontrolelrs.							

LPC1000/2000 FLASH MEMORY ORGANIZATION

LPC1/2/4000 microcontrollers come with several sizes of the on-chip Flash memory: 8, 16, 32, 48, 64, 80, 96, 128, 256, 512 or 1024 kilobytes (kB). The Flash memory system contains from 2 to 32 sectors, varying in size from 4 kB to 64 kB. Flash memory begins at address 0x0000 0000h (LPC11/13/15/1700/2000/40xx) or 0x1A00 0000h (LPC181x/2x/3x/5x and LPC431x/2x/3x/5x) and continues upward. In some of the LPC1/2/4000 microcontrollers not all of Flash sectors are available for write access. Table 17 thru Table 23 show available Flash memory for every LPC1/2/4000 microcontroller covered with this programming specification.

Initially, access to the Flash memory is possible via the serial bootloader only. The parallel bootloader has to be loaded to the device through a serial link. The whole process is described in the following chapters.

Table 17: LPC2000 Flash Memory Sectors and Addresses accessible for parallel programming (LPC2104/2105/2106/2109/2114/2119/2212 and LPC2124/2129/2194/2214/2292/2294)

Sector Number (dec/hex)	LPC2109 (64 kB of on-chip Flash available for user's application)		LPC2104/2105/2106/2114/2119/2212 (120 kB of on-chip Flash available for user's application)		LPC2124/2129/2194/2214/2292/2294 (248 kB of on-chip Flash available for user's application)	
	Start - End Address (hex)	Sector size (kB)	Start - End Address (hex)	Sector size (kB)	Start - End Address (hex)	Sector size (kB)
0 / 0x00	0000 0000 - 0000 1FFF	8	0000 0000 - 0000 1FFF	8	0000 0000 0000 1FFF	8
1 / 0x01	0000 2000 - 0000 3FFF	8	0000 2000 - 0000 3FFF	8	0000 2000 - 0000 3FFF	8
2 / 0x02	0000 4000 - 0000 5FFF	8	0000 4000 - 0000 5FFF	8	0000 4000 - 0000 5FFF	8
3 / 0x03	0000 6000 - 0000 7FFF	8	0000 6000 - 0000 7FFF	8	0000 6000 - 0000 7FFF	8
4 / 0x04	0000 8000 - 0000 9FFF	8	0000 8000 - 0000 9FFF	8	0000 8000 - 0000 9FFF	8
5 / 0x05	0000 A000 - 0000 BFFF	8	0000 A000 - 0000 BFFF	8	0000 A000 - 0000 BFFF	8
6 / 0x06	0000 C000 - 0000 DFFF	8	0000 C000 - 0000 DFFF	8	0000 C000 - 0000 DFFF	8
7 / 0x07	0000 E000 - 0000 FFFF	8	0000 E000 - 0000 FFFF	8	0000 E000 - 0000 FFFF	8
8 / 0x08	Not available		0001 0000 - 0001 1FFF	8	0001 0000 - 0001 FFFF	64
9 / 0x09			0001 2000 - 0001 3FFF	8	0002 2000 - 0002 FFFF	64
10 / 0x0A			0001 4000 - 0001 5FFF	8	0003 0000 - 0003 1FFF	8
11 / 0x0B			0001 6000 - 0001 7FFF	8	0003 2000 - 0003 3FFF	8
12 / 0x0C			0001 8000 - 0001 9FFF	8	0003 4000 - 0000 5FFF	8
13 / 0x0D			0001 A000 - 0001 BFFF	8	0003 6000 - 0003 7FFF	8
14 / 0x0E			0001 C000 - 0001 DFFF	8	0003 8000 - 0003 9FFF	8
15 / 0x0F			Not available		0003 A000 - 0003 BFFF	8
16 / 0x10					0003 C000 - 0003 DFFF	8

Table 18: LPC2000 Flash Memory Sectors and Addresses accessible for parallel programming (LPC2101/2/3, LPC2131/2/4/6/8, LPC2141/2/4/6/8, LPC2361/62/64/65/66/68/77/78/87/88, LPC2458/68/78)

Sector Number (dec/hex)	Start - End Address (hex)	Sector size (kB)	Part number and amount of the on-chip Flash memory ready for user's application (+ indicates available sector)					
			LPC2101/2/3 LPC2131/2141 (32 kB)	LPC2132/2142 LPC2361 (64 kB)	LPC2134/2144 LPC2362/64 (128 kB)	LPC2136/2146 LPC2365/66 (256 kB)	LPC2138/2148 (500 kB)	LPC2367/68/77/78 2387/88/2458/68/78 (504 kB)
0 / 0x00	0000 0000 - 0000 0FFF	4	+	+	+	+	+	+
1 / 0x01	0000 1000 - 0000 1FFF	4	+	+	+	+	+	+
2 / 0x02	0000 2000 - 0000 2FFF	4	+	+	+	+	+	+
3 / 0x03	0000 3000 - 0000 3FFF	4	+	+	+	+	+	+
4 / 0x04	0000 4000 - 0000 4FFF	4	+	+	+	+	+	+
5 / 0x05	0000 5000 - 0000 5FFF	4	+	+	+	+	+	+
6 / 0x06	0000 6000 - 0000 6FFF	4	+	+	+	+	+	+
7 / 0x07	0000 7000 - 0000 7FFF	4	+	+	+	+	+	+
8 / 0x08	0000 8000 - 0000 FFFF	32	NA	+	+	+	+	+
9 / 0x09	0001 0000 - 0001 7FFF	32	Not Available		+	+	+	+
10 / 0x0A	0001 8000 - 0001 FFFF	32			+	+	+	+
11 / 0x0B	0002 0000 - 0002 7FFF	32	Not Available			+	+	+
12 / 0x0C	0002 8000 - 0002 FFFF	32				+	+	+
13 / 0x0D	0003 0000 - 0003 7FFF	32				+	+	+
14 / 0x0E	0003 8000 - 0003 FFFF	32				+	+	+
15 / 0x0F	0004 0000 - 0004 7FFF	32	Not Available				+	+
16 / 0x10	0004 8000 - 0004 FFFF	32					+	+
17 / 0x11	0005 0000 - 0005 7FFF	32					+	+
18 / 0x12	0005 8000 - 0005 FFFF	32					+	+
19 / 0x13	0006 0000 - 0006 7FFF	32					+	+
20 / 0x14	0006 8000 - 0006 FFFF	32					+	+
21 / 0x15	0007 0000 - 0007 7FFF	32					+	+
22 / 0x16	0007 8000 - 0007 8FFF	4					+	+
23 / 0x17	0007 9000 - 0007 9FFF	4					+	+
24 / 0x18	0007 A000 - 0007 AFFF	4					+	+
25 / 0x19	0007 B000 - 0007 BFFF	4					+	+
26 / 0x1A	0007 C000 - 0007 CFFF	4					+	+
27 / 0x1B	0007 D000 - 0007 DFFF	4					NA	+

Table 19: LPC18xx and LPC43xx Flash Memory Sectors and Addresses accessible for parallel programming (LPC1812/13/15/17/22/23/25/27/33/37/53/57 and LPC4312/13/15/17/22/23/25/27/33/37/53/57)

Flash bank	Sector Number (dec/hex)	Start - End Address (hex)		Part number and amount of the on-chip Flash memory ready for user's application (+ indicates available sector)			
			Sector size (kB)	LPC18x2 LPC43x2 (512 kB)	LPC18x3 LPC43x3 (512 kB)	LPC18x5 LPC43x5 (768 kB)	LPC18x7 LPC43x7 (1024 kB)
0	0 / 0x00	1A00 0000 - 1A00 1FFF	8	+	+	+	+
	1 / 0x01	1A00 2000 - 1A00 3FFF	8	+	+	+	+
	2 / 0x02	1A00 4000 - 1A00 5FFF	8	+	+	+	+
	3 / 0x03	1A00 6000 - 1A00 7FFF	8	+	+	+	+
	4 / 0x04	1A00 8000 - 1A00 9FFF	8	+	+	+	+
	5 / 0x05	1A00 A000 - 1A00 BFFF	8	+	+	+	+
	6 / 0x06	1A00 C000 - 1A00 DFFF	8	+	+	+	+
	7 / 0x07	1A00 E000 - 1A00 FFFF	8	+	+	+	+
	8 / 0x08	1A01 0000 - 1A01 FFFF	64	+	+	+	+
	9 / 0x09	1A02 0000 - 1A02 FFFF	64	+	+	+	+
	10 / 0x0A	1A03 0000 - 1A03 FFFF	64	+	+	+	+
	11 / 0x0B	1A04 0000 - 1A04 FFFF	64	+	NA	+	+
	12 / 0x0C	1A05 0000 - 1A05 FFFF	64	+		+	+
	13 / 0x0D	1A06 0000 - 1A06 FFFF	64	+		NA	+
	14 / 0x0E	1A07 0000 - 1A07 FFFF	64	+			+
1	0 / 0x00	1B00 0000 - 1B00 1FFF	8	NA	+	+	+
	1 / 0x01	1B00 2000 - 1B00 3FFF	8		+	+	+
	2 / 0x02	1B00 4000 - 1B00 5FFF	8		+	+	+
	3 / 0x03	1B00 6000 - 1B00 7FFF	8		+	+	+
	4 / 0x04	1B00 8000 - 1B00 9FFF	8		+	+	+
	5 / 0x05	1B00 A000 - 1B00 BFFF	8		+	+	+
	6 / 0x06	1B00 C000 - 1B00 DFFF	8		+	+	+
	7 / 0x07	1B00 E000 - 1B00 FFFF	8		+	+	+
	8 / 0x08	1B01 0000 - 1B01 FFFF	64		+	+	+
	9 / 0x09	1B02 0000 - 1B02 FFFF	64		+	+	+
	10 / 0x0A	1B03 0000 - 1B03 FFFF	64		+	+	+
	11 / 0x0B	1B04 0000 - 1B04 FFFF	64		NA	+	+
	12 / 0x0C	1B05 0000 - 1B05 FFFF	64			+	+
	13 / 0x0D	1B06 0000 - 1B06 FFFF	64			NA	+
	14 / 0x0E	1B07 0000 - 1B07 FFFF	64				+

Table 20: LPC177x/8x and LPC407x/8x Flash Memory Sectors and Addresses accessible for parallel programming (LPC1774/76/77/78/85/86/87/88 and LPC4072/74/76/78/88)

Sector Number (dec/hex)	Start - End Address (hex)	Sector size (kB)	Part number and amount of the on-chip Flash memory ready for user's application (+ indicates available sector)			
			LPC1772 LPC4072 (64 kB)	LPC1774 LPC4074 (128 kB)	LPC1776/ LPC1785/86 LPC4076 (256 kB)	LPC1777/78 LPC1787/88 LPC4078/88 (512 kB)
0 / 0x00	0000 0000 - 0000 0FFF	4	+	+	+	+
1 / 0x01	0000 1000 - 0000 1FFF	4	+	+	+	+
2 / 0x02	0000 2000 - 0000 2FFF	4	+	+	+	+
3 / 0x03	0000 3000 - 0000 3FFF	4	+	+	+	+
4 / 0x04	0000 4000 - 0000 4FFF	4	+	+	+	+
5 / 0x05	0000 5000 - 0000 5FFF	4	+	+	+	+
6 / 0x06	0000 6000 - 0000 6FFF	4	+	+	+	+
7 / 0x07	0000 7000 - 0000 7FFF	4	+	+	+	+
8 / 0x08	0000 8000 - 0000 8FFF	4	+	+	+	+
9 / 0x09	0000 9000 - 0000 9FFF	4	+	+	+	+
10 / 0x0A	0000 A000 - 0000 AFFF	4	+	+	+	+
11 / 0x0B	0000 B000 - 0000 BFFF	4	+	+	+	+
12 / 0x0C	0000 C000 - 0000 CFFF	4	+	+	+	+
13 / 0x0D	0000 D000 - 0000 DFFF	4	+	+	+	+
14 / 0x0E	0000 E000 - 0000 EFFF	4	+	+	+	+
15 / 0x0F	0000 F000 - 0000 FFFF	4	+	+	+	+
16 / 0x08	0001 0000 - 0001 7FFF	32	Not Available	+	+	+
17 / 0x09	0001 8000 - 0001 FFFF	32		+	+	+
18 / 0x0A	0002 0000 - 0002 7FFF	32	Not Available		+	+
19 / 0x0B	0002 8000 - 0002 FFFF	32			+	+
20 / 0x0C	0003 0000 - 0003 7FFF	32			+	+
21 / 0x0D	0003 8000 - 0003 FFFF	32			+	+
22 / 0x0E	0004 0000 - 0004 7FFF	32	Not Available			+
23 / 0x0F	0004 8000 - 0004 FFFF	32				+
24 / 0x10	0005 0000 - 0005 7FFF	32				+
25 / 0x11	0005 8000 - 0005 FFFF	32				+
26 / 0x12	0006 0000 - 0006 7FFF	32				+
27 / 0x13	0006 8000 - 0006 FFFF	32				+
28 / 0x14	0007 0000 - 0007 7FFF	32				+
29 / 0x15	0007 8000 - 0007 8FFF	32				+

Table 21: LPC175x/6x Flash Memory Sectors and Addresses accessible for parallel programming (LPC1751/52/54/56/58/59/64/65/66/67/68/69)

Sector Number (dec/hex)	Start - End Address (hex)	Sector size (kB)	Part number and amount of the on-chip Flash memory ready for user's application (+ indicates available sector)				
			LPC1751 (32 kB)	LPC1752 (64 kB)	LPC1754/1764 (128 kB)	LPC1756 LPC1763/65/66 (256 kB)	LPC1758/59 LPC1767/68/69 (512 kB)
0 / 0x00	0000 0000 - 0000 0FFF	4	+	+	+	+	+
1 / 0x01	0000 1000 - 0000 1FFF	4	+	+	+	+	+
2 / 0x02	0000 2000 - 0000 2FFF	4	+	+	+	+	+
3 / 0x03	0000 3000 - 0000 3FFF	4	+	+	+	+	+
4 / 0x04	0000 4000 - 0000 4FFF	4	+	+	+	+	+
5 / 0x05	0000 5000 - 0000 5FFF	4	+	+	+	+	+
6 / 0x06	0000 6000 - 0000 6FFF	4	+	+	+	+	+
7 / 0x07	0000 7000 - 0000 7FFF	4	+	+	+	+	+
8 / 0x08	0000 8000 - 0000 8FFF	4	NA	+	+	+	+
9 / 0x09	0000 9000 - 0000 9FFF	4		+	+	+	+
10 / 0x0A	0000 A000 - 0000 AFFF	4		+	+	+	+
11 / 0x0B	0000 B000 - 0000 BFFF	4		+	+	+	+
12 / 0x0C	0000 C000 - 0000 CFFF	4		+	+	+	+
13 / 0x0D	0000 D000 - 0000 DFFF	4		+	+	+	+
14 / 0x0E	0000 E000 - 0000 EFFF	4		+	+	+	+
15 / 0x0F	0000 F000 - 0000 FFFF	4		+	+	+	+
16 / 0x08	0001 0000 - 0001 7FFF	32	Not Available		+	+	+
17 / 0x09	0001 8000 - 0001 FFFF	32			+	+	+
18 / 0x0A	0002 0000 - 0002 7FFF	32	Not Available			+	+
19 / 0x0B	0002 8000 - 0002 FFFF	32				+	+
20 / 0x0C	0003 0000 - 0003 7FFF	32				+	+
21 / 0x0D	0003 8000 - 0003 FFFF	32				+	+
22 / 0x0E	0004 0000 - 0004 7FFF	32	Not Available				+
23 / 0x0F	0004 8000 - 0004 FFFF	32					+
24 / 0x10	0005 0000 - 0005 7FFF	32					+
25 / 0x11	0005 8000 - 0005 FFFF	32					+
26 / 0x12	0006 0000 - 0006 7FFF	32					+
27 / 0x13	0006 8000 - 0006 FFFF	32					+
28 / 0x14	0007 0000 - 0007 7FFF	32					+
29 / 0x15	0007 8000 - 0007 FFFF	32					+

Table 22: LPC1224/1225/1226/1227 Flash Memory Sectors and Addresses accessible for parallel programming (all LPC1224/5/6/7 and LPC12D27 flash sectors are 4 kB in size)

<table>
<tr><th rowspan="2">Sector Number

(dec/hex)</th><th rowspan="2">Start - End Address (hex)</th><th></th><th colspan="6">Part number and amount of the on-chip Flash memory ready for user's application (+ indicates available sector)</th></tr>
<tr><th>Sector size (kB)</th><th>LPC1224/101 (32 kB)</th><th>LPC1224/121 (48 kB)</th><th>LPC1225/301 (64 kB)</th><th>LPC1225/321 (80 kB)</th><th>LPC1226/301 (96 kB)</th><th>LPC1227/301 LPC12D27/301 (128 kB)</th></tr>
<tr><td>0 / 0x00</td><td>0000 0000 - 0000 0FFF</td><td>4</td><td>+</td><td>+</td><td>+</td><td>+</td><td>+</td><td>+</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>7 / 0x07</td><td>0000 7000 - 0000 7FFF</td><td>4</td><td>+</td><td>+</td><td>+</td><td>+</td><td>+</td><td>+</td></tr>
<tr><td>8 / 0x08</td><td>0000 8000 - 0000 8FFF</td><td>4</td><td rowspan="3">NA</td><td>+</td><td>+</td><td>+</td><td>+</td><td>+</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>11 / 0x0B</td><td>0000 B000 - 0000 BFFF</td><td>4</td><td>+</td><td>+</td><td>+</td><td>+</td><td>+</td></tr>
<tr><td>12 / 0x0C</td><td>0000 C000 - 0000 FFFF</td><td>4</td><td colspan="2" rowspan="3">Not Available</td><td>+</td><td>+</td><td>+</td><td>+</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>15 / 0x0F</td><td>0000 F000 - 0000 FFFF</td><td>4</td><td>+</td><td>+</td><td>+</td><td>+</td></tr>
<tr><td>16 / 0x10</td><td>0001 0000 - 0001 0FFF</td><td>4</td><td colspan="3" rowspan="3">Not Available</td><td>+</td><td>+</td><td>+</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>19 / 0x13</td><td>0001 3000 - 0001 3FFF</td><td>4</td><td>+</td><td>+</td><td>+</td></tr>
<tr><td>20 / 0x14</td><td>0001 4000 - 0001 4FFF</td><td>4</td><td colspan="4" rowspan="3">Not Available</td><td>+</td><td>+</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>23 / 0x17</td><td>0001 7000 - 0001 7FFF</td><td>4</td><td>+</td><td>+</td></tr>
<tr><td>24 / 0x18</td><td>0001 8000 - 0001 8FFF</td><td>4</td><td colspan="5" rowspan="3">Not Available</td><td>+</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>31 / 0x1F</td><td>0001 F000 - 0001 FFFF</td><td>4</td><td>+</td></tr>
</table>

Table 23: LPC1100/1300 Flash Memory Sectors and Addresses accessible for parallel programming (LPC1111/12/13/14, LPC1101LV/02LV/12LV/14LV, LPC11A02/A04/A11/A12/A13/A14, LPC11C12/C14/C22/C24, LPC11D14, LPC11E11/E12/E13/E14, LPC11U12/U13/U14/U23/U24/U34/U35/U36/U37 and LPC1311/13/15/16/17/42/43/45/46/47)

Sector Number (dec/ hex)	Start - End Address (hex)	Sector size (kB)	Part number and amount of the on-chip Flash memory ready for user's application (+ indicates available sector)									
			LPC1111, LPC11A11, LPC11E11, LPC1311, LPC1311/01 (8 kB)	LPC1112, LPC1112LV, LPC11A02, LPC11A12, LPC11E12,LPC11C12, LPC11C22,LPC11U12, LPC1342 (16 kB)	LPC1113, LPC11A13, LPC11E13, LPC11U13, LPC11U23 (24 kB)	LPC1114, LPC1101LV/02LV/14LV, LPC11A04, LPC11A14, LPC11E14, LPC11C14, LPC11C24, LPC11D14, LPC11U14, LPC11U24LPC1313/1343, LPC1313/01, LPC1315/1345 (32 kB)	LPC11U34 (40 kB)	LPC11U34, LPC1114/323, LPC1316/1346 (48 kB)	LPC1114/323 (56 kB)	LPC11U35, LPC1115/303, LPC1317/1347 (64 kB)	LPC11U36 (96 kB)	LPC11U37 (128 kB)
0 / 0x00	0000 0000 - 0000 0FFF	4	+	+	+	+	+	+	+	+	+	+
1 / 0x01	0000 1000 - 0000 1FFF	4	+	+	+	+	+	+	+	+	+	+
2 / 0x02	0000 2000 - 0000 2FFF	4	NA	+	+	+	+	+	+	+	+	+
3 / 0x03	0000 3000 - 0000 3FFF	4		+	+	+	+	+	+	+	+	+
4 / 0x04	0000 4000 - 0000 4FFF	4	Not Available		+	+	+	+	+	+	+	+
5 / 0x05	0000 5000 - 0000 5FFF	4			+	+	+	+	+	+	+	+
6 / 0x06	0000 6000 - 0000 6FFF	4	Not Available			+	+	+	+	+	+	+
7 / 0x07	0000 7000 - 0000 7FFF	4				+	+	+	+	+	+	+
8 / 0x08	0000 8000 - 0000 8FFF	4				Not Available	+	+	+	+	+	+
9 / 0x09	0000 9000 - 0000 9FFF	4					+	+	+	+	+	+
10/0x0A	0000 A000 - 0000 AFFF	4					NA	+	+	+	+	+
11/0x0B	0000 B000 - 0000 BFFF	4						+	+	+	+	+
12/0x0C	0000 C000 - 0000 CFFF	4						NA	+	+	+	+
13/0x0D	0000 D000 - 0000 DFFF	4							+	+	+	+
14/0x0E	0000 E000 - 0000 EFFF	4							NA	+	+	+
15/0x0F	0000 F000 - 0000 FFFF	4								+	+	+
16/0x10 ... 23/0x17	0001 0000 - 0x0001 1FFF ... 0x0001 7000 - 0x0001 7FFF	4 ... 4								NA	+	+
24/0x18 ... 31/0x1F	0001 8000 - 0x0001 8FFF ... 0x0001 F000 - 0x0001 FFFF	4 ... 4									NA	+

PARALLEL LPC1/2/4000 FLASH PROGRAMMING

When delivered to the customer all LPC1/2/4000 microcontrollers are equipped with serial download capabilities only. Therefore parallel LPC1/2/4000 Flash programming solution uses the UART0 to upload a small programming application into the microcontroller. Once executed, this application will be used as an interface between the programming hardware (the parallel programmer) and the programmed chip. For more details on the LPC1/2/4000 on-board Flash memory system and programming, please see the specific LPC1/2/4000 part dedicated User Manual.

Entering Flash Programming and Verification Mode

External clock signal of 10 MHz must be available to the programmed part on pins X1/2 for the whole time of its programming/operation.

After power is applied to the microcontroller, reset of the device follows. It is advised that 3.3V does not come before 1.8 power supply (when 1.8V power supply is needed), since 1.8V powers the core and 3.3V enables pins to operate properly. After a low level is applied to the Reset pin, the DBGSEL line (available in LPC2101/2/3/4/5/6 only) must go low, as well as the ISP line. Reset signal has to stay low for a period of 10 milliseconds. After this time, Reset signal goes high and stays high for the rest of programming and verification process.

The ISP line must be held low for additional 5 milliseconds after the Reset line has gone high. After this delay expires, the ISP line might be used as a data line (D6) in case one pin is used to host both the ISP and D6 function or it might be used as a Command (CMD) line.

Once the DBGSEL line becomes low, it stays low for the rest of the device programming.

Important: not all LPC1/2/4000 devices need multiplexed ISP/(D6/CMD) lines. See Table 1 thru Table 16 for details.

Some LPC1100 and LPC1300 devices need an additional line to be controlled while the ISP line is driven LOW when entering the Flash programming mode. See Table 5 thru Table 8 for details.

It is important to remember that LPC11xxLV microcontrollers are 1.8V only devices! If either a power supply or any other signal exceeds this limit, the programmed device might be permanently damaged!

Initial data load Using UART0

At this point, the ISP (In System Programming) software module residing in the LPC1/2/4000 microcontroller has been activated. The next step requires serial data load into the device.

The ISP communicates over the UART0 peripheral. Hence, the ISP serial data load uses UART based communication at 9600 bit/s, 1 start, 8 data, 1 stop bit format without parity or hardware/software handshake. The LPC1/2/4000 receives data on the SDIN pin. Echoed data as well as additional content is available on the SDOUT pin. While data on the SDOUT can be used for process monitoring, they are not crucial for flash programming and verification.

Several messages have to be sent from the programmer to the LPC1/2/4000 device in this phase of programming. After the last byte of each message is sent over the SDIN line, a specific delay is required before the next message is transmitted to the SDIN. The order of messages, their contents and the delay after each of them are listed in Table 25 (LPC23xx/24xx), Table 24 (from LPC21xx/22xx), Table 26 (LPC18xx and LPC43xx), Table 27 (LPC177x/8x and LPC407x/8x), Table 28 (LPC175x/6x), Table 29 (LPC1311/13/42/43), Table 30 (LPC12xx/12Dxx), Table 31 (LPC111x/11Cxx/11Dxx), Table 32 (LPC11Uxx/LPC11Exx/1315/16/17/45/46/47), Table 33 (LPC11Axx), Table 34 (20-pin LPC1111/1112, LPC11xxLV), and Table 35 (24-pin LPC1112).

Message content is case sensitive. After echoing every message on the SDOUT pin, the LPC1000/2000 the microcontroller will report if a received message is recognized as a valid one, and if so, execute it. List of error codes is available in the appendix of this document.

Although the 9600 bit/s baudrate is suggested in the text above, the LPC1/2/4000 is capable of establishing serial communication of 19200 and 38400 bit/s when external crystal of 10 MHz is used, too. However, this specification covers waveforms and timing related to the 9600 bit/s option only.

Table 24: Messages required to be sent over the serial channel (LPC21xx/LPC22xx devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'10000'+<CR>+<LF>[1] Note: this message informs the microcontroller that its external clock is 10 MHz	'10000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 1073742336 720'+<CR>+<LF>[1] Note: this command loads 720 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x4000 0200 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 1073742336 720' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 1073743036 A'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x4000 04BC	'G 1073743036 A'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

.

Table 25: Messages required to be sent over the serial channel (LPC23xx/24xx devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'10000'+<CR>+<LF>[1] Note: this message informs the microcontroller that its external clock is 10 MHz	'10000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 1073742336 1076'+<CR>+<LF>[1] Note: this command loads 1076 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x4000 0200 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 1073742336 1076' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 1073742336 A'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x4000 0200	'G 1073742336 A'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

Table 26: Messages required to be sent over the serial channel (LPC18xx and LPC43xx devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'96000'+<CR>+<LF>[1] Note: this message informs the microcontroller the operating clock is 96 MHz	'96000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 268436544 568'+<CR>+<LF>[1] Note: this command loads 568 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x1000 0440 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 268436544 568' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 268436544 T'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x1000 0440	'G 268436544 T'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

Table 27: Messages required to be sent over the serial channel (LPC177x/8x and LPC407x/8x devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'10000'+<CR>+<LF>[1] Note: this message informs the microcontroller that its external clock is 10 MHz	'10000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 268436480 680'+<CR>+<LF>[1] Note: this command loads 680 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x1000 0400 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 268436480 680' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 268436492 T'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x1000 040C	'G 268436492 T'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

Table 28: Messages required to be sent over the serial channel (LPC175x/6x devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'10000'+<CR>+<LF>[1] Note: this message informs the microcontroller that its external clock is 10 MHz	'10000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 268436256 1152'+<CR>+<LF>[1] Note: this command loads 1152 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x1000 0320 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 268436256 1152' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 268436616 T'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x1000 0488	'G 268436616 T'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

Table 29: Messages required to be sent over the serial channel (LPC1311/13/42/43 devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'10000'+<CR>+<LF>[1] Note: this message informs the microcontroller that its external clock is 10 MHz	'10000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 268436256 1220'+<CR>+<LF>[1] Note: this command loads 1220 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x1000 0320 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 268436256 1220' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 268437300 T'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x1000 0734	'G 268437300 T'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

Table 30: Messages required to be sent over the serial channel (LPC12xx/12Dxx devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'10000'+<CR>+<LF>[1] Note: this message informs the microcontroller that its external clock is 10 MHz	'10000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 268436544 636'+<CR>+<LF>[1] Note: this command loads 636 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x1000 0440 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 268436544 636' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 268436544 T'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x1000 0440	'G 268436544 T'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

Table 31: Messages required to be sent over the serial channel
(LPC1111/12/13/14, LPC11C12/C14/C22/C24 and LPC11D14 devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'10000'+<CR>+<LF>[1] Note: this message informs the microcontroller that its external clock is 10 MHz	'10000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 268436544 664'+<CR>+<LF>[1] Note: this command loads 664 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x1000 0440 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 268436544 664' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 268436544 T'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x1000 0440	'G 268436544 T'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

Table 32: Messages required to be sent over the serial channel (LPC11U12/U13/U14/U23/U24/U34/U35/U36/U37, LPC11E11/E12/E13/E14 and LPC1315/16/17/45/46/47 devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'10000'+<CR>+<LF>[1] Note: this message informs the microcontroller that its external clock is 10 MHz	'10000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 268436544 608'+<CR>+<LF>[1] Note: this command loads 608 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x1000 0440 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 268436544 608' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 268436544 T'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x1000 0440	'G 268436544 T'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

Table 33: Messages required to be sent over the serial channel (LPC11A02/A04/A11/A12/A13/A14 devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'10000'+<CR>+<LF>[1] Note: this message sent for legacy reasons sinceLPC11Axx devices do not need an external clock	'10000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 268436544 588'+<CR>+<LF>[1] Note: this command loads 588 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x1000 0440 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 268436544 588' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 268436544 T'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x1000 0440	'G 268436544 T'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

Table 34: Messages required to be sent over the serial channel (20-pin LPC1111/1112 and LPC1101LV/1102LV/1112LV/1114LV devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'10000'+<CR>+<LF>[1] Note: this message sent for legacy reasons sinceLPC11Axx devices do not need an external clock	'10000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 268436544 684'+<CR>+<LF>[1] Note: this command loads 684 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x1000 0440 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 268436544 684' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 268436544 T'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x1000 0440	'G 268436544 T'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

Table 35: Messages required to be sent over the serial channel (24-pin LPC1112 devices)

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'12000'+<CR>+<LF>[1] Note: this message has no effect on a 20-pin LPC1112 device since it uses an internal 12 MHz oscillator during parallel programming	'12000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'W 268436544 644'+<CR>+<LF>[1] Note: this command loads 644 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x1000 0440 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 268436544 644' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
6	'G 268436544 T'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x1000 0440	'G 268436544 T'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

After message number 5 was sent to the device and the required delay of 12 ms expired, a sequence of UUENCODED data has to be transmitted over the SDIN line. This sequence consists of 17 lines (LPC21xx/22xx), 26 lines (LPC23xx/24xx), 17 lines (LPC407x/8x, LPC177x/8x, 20-pin LPC1111/1112, LPC11xxLV), 28 lines (LPC175x/6x), 30 lines (LPC1311/13/42/43), 16 lines (LPC12xx/LPC111x/LPC11Cxx, 24-pin LPC1112) or 15 lines (LPC11Uxx/LPC11Exx/LPC11Axx/LPC1315/16/17/45/46/47) each line containing up to 61 characters and ending with <CR><LF>. After every of these lines is sent over the SDIN, a delay of 10 ms must be added before the next line of UUENCODED data is transmitted.

In case of LPC21xx/22xx, once the checksum line is sent after the 16th line of data, a delay of 20 ms must be implemented before the programmer continues with the message number 6 from Table 24. Full listing of the UUENCODED data is in the appendix section of this document.

In case of LPC23xx/24xx, once the checksum line is sent after the 20th line of data, a delay of 20 ms must be implemented before the programmer continues with the remaining 4 lines and the corresponding checksum line. After the second checksum is sent and a delay of 20 ms is made, message number 6 from Table 25 can follow. Full listing of the UUENCODED data is in the appendix section of this document.

In case of LPC18xx and LPC43xx, once the checksum line is sent after the 13th line of data, a delay of 20 ms must be implemented before the programmer continues with the message number 6 from Table 26. Full listing of the UUENCODED data is in the appendix section of this document.

In case of LPC407x/8x, LPC177x/8x, 20-pin LPC1111/1112, or LPC11xxLV once the checksum line is sent after the 16th line of data, a delay of 20 ms must be implemented before the programmer continues with the message number 6 from Table 27/Table 34. Full listing of the UUENCODED data is in the appendix section of this document.

In case of LPC175x/6x, once the checksum line is sent after the 20th line of data, a delay of 20 ms must be implemented before the programmer continues with the remaining 6 lines and the corresponding checksum line. After the second checksum is sent and a delay of 20 ms is made, message number 6 from Table 28 can follow. Full listing of the UUENCODED data is in the appendix section of this document.

In case of LPC1311/13/42/43, once the checksum line is sent after the 20th line of data, a delay of 20 ms must be implemented before the programmer continues with the remaining 8 lines and the corresponding checksum line. After the second checksum is sent and a delay of 20 ms is made, message number 6 from Table 29 can follow. Full listing of the UUENCODED data is in the appendix section of this document.

In case of LPC12xx/12Dxx/111x/11Cxx/11Dxx/24-pin LPC1112, once the checksum line is sent after the 15th line of data, a delay of 20 ms must be implemented before the programmer continues with the message number 6 from Table 30/Table 31/Table 35. Full listing of the UUENCODED data is in the appendix section of this document.

In case of LPC11Uxx/11Exx/11Axx/LPC1315/16/17/45/46/47, once the checksum line is sent after the 14th line of data, a delay of 20 ms must be implemented before the programmer continues with the message number 6 from Table 32/Table 33. Full listing of the UUENCODED data is in the appendix section of this document.

Only when the full set of UUENCODED data plus the checksum(s) are loaded into the microcontroller and confirmed by the LPC1/2/4000 as correctly received, message 6 can be sent. If there were no errors in the data transfer, after no more then 10 ms as message 6 has been sent, the Status lines (S[3:0]) together with the RES line will make a transition from high to low level. If there is no such transition, an error occurred in the data transfer and the programmed device has to go through the Reset sequence (previously described), followed by a complete procedure of entering Flash programming and verification mode. When Status lines switch to low level, the SDOUT line will change its function to the CMD (Command Line indicator) and this pin on the LPC2000 will become an input pin. The SDIN line changes into the RES line (Response Line indicator) and this pin on the programmed device becomes an output pin.

LPC4000, LPC23xx/24xx, LPC17xx, LPC13xx, LPC12xx, and LPC11xx parallel programming solutions have dedicated (non-multiplexed) SDIN/SDOUT/CMD/RES lines as outlined from Table 2 to Table 9 and Table 12 to Table 16.

Important: Due to low pincount 20-pin LPC1112 and LPC11xxLV devices have some control signals multiplexed as specified in Table 10 and Table 11. Because of the same reason the 20-pin and 24-pin LPC1112 as well as LPC11xxLV devices do not provide Status lines.

PARALLEL DATA TRANSFER (TWO-WAY HANDSHAKE)

After correct data have been loaded and confirmed by the LPC1/2/4000 device (all four Status lines and the RES line performing a high to low transition), the only mechanism of communication between the programmer and the programmed device will be a parallel data transfer with two-way handshake.

Using the parallel communication mechanism, a programmer can perform one of several activities:

- Read Part ID
- Read Boot Code Version
- Read Memory
- Erase Flash Memory (single sector or full chip)
- Blank Check (full chip only)
- Program Flash Memory and Verify

This parallel data transfer implementation uses 8 data lines (D[7:0]), the HSP (handshake generated by the programmer) line, the HSC (handshake generated by the LPC1/2/4000) line, as well as the CMD (Command) and the RES (Response) lines. Status of every command execution can be checked on the Status S[3:0] lines.

There are two types of communication sequences that are allowed in communication between the programmer and the LPC1/2/4000 device. The first one is when the programmer sends a command to the device. The second one is when the programmer reads data from the device. Detailed waveforms showing control and data signals for both cases are shown in Figure 83 and Figure 84.

The parallel programmer is solely responsible for the command and read sequence initialization. The programmed LPC1/2/4000 device is only capable of responding to requests sent by the programmer. The programmed device can not at any point in time send any data on its own initiative.

Software running on the LPC1/2/4000 has no time-out features of any kind. It is up to the parallel programmer to verify proper waveform and signal integrity during the programming procedure and to reset the programmed LPC1/2/4000 part in case it becomes necessary.

Command Sequence Issued by the Programmer

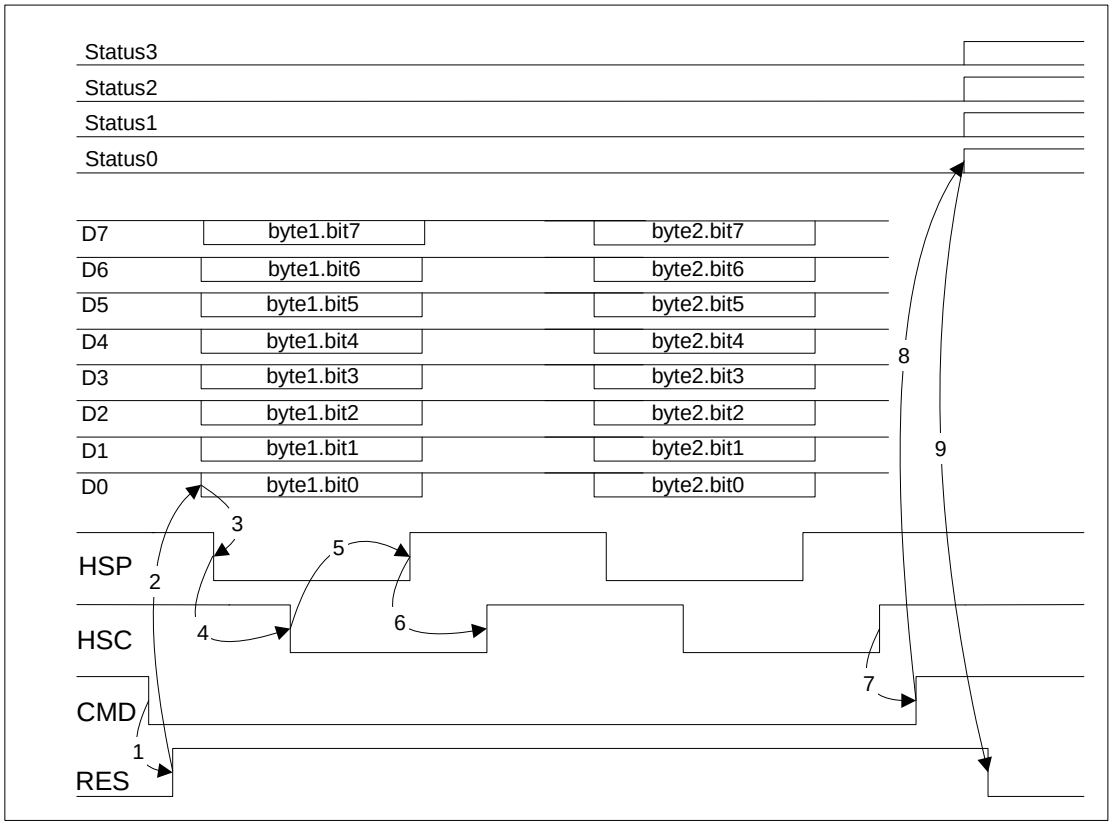


Figure 83: Command waveform (case of a two byte command transmission)

During the command sequence, only the programmer outputs data on the data bus (D[7:0]).

Before the programmer sends a command, the HSP, HSC and CMD line must be high, while the RES line must be low.

The parallel programmer initiates command generating a high-to-low transition on the CMD line. Within the next 5 microseconds the LPC1/2/4000 microcontroller will generate a low-to-high transition on the RES line (sequence 1, Figure 83). A high level on the RES line indicates that the LPC1/2/4000 is ready to accept new instructions from the parallel programmer.

The programmer outputs data on the data lines D[7:0] (seq. 2). When data are available on the data bus, the programmer generates a high-to-low transition on the HSP line (seq. 3). The programmed microcontroller reads data from the parallel bus and within the next 5 microseconds generates a high-to-low transition on the HSC line (seq. 4). The programmer drives the HSP line high (seq. 5) and the microcontroller within the next 5 microseconds drives the HSC line high, too (seq. 6). At this point, new data can be set on the data bus, and sequence 3-4-5-6 can be repeated.

When the programmer has sent all data and the LPC1/2/4000 has confirmed them (3-4-5-6 cycle), the programmer generates a low-to-high transition on the CMD line (seq. 7). Regardless of the character of command that was just sent, the programmed device MUST update Status lines (seq. 8), and notify the programmer with a high-to-low transition on the RES line that the received command has been executed (seq. 9). A delay between the rising edge on the CMD and the falling edge on the RES line in sequence 9 is command dependent.

Read Sequence Issued by the Programmer

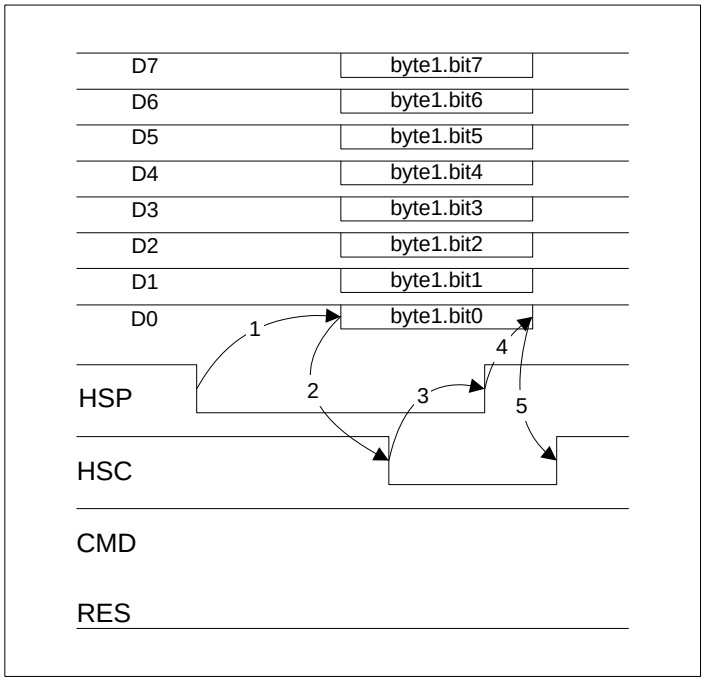


Figure 84: Data Read waveform (case of a single byte read transmission)

During a read sequence only the LPC1/2/4000 outputs data on the data bus (D[7:0]).

In order to perform a read operation, a proper command must be given by the parallel programmer to the programmed LPC1/2/4000 device. While the programmer is reading data from the LPC1/2/4000 microcontroller, the CMD line stays high and the RES line stays low.

By generating a high-to-low transition on the HSP line, the parallel programmer initiates a read cycle. The LPC1/2/4000 microcontroller puts data on the data bus (sequence 1, Figure 84), and within next 7 microseconds generates a high-to-low transition on the HSC line (sequence 2). After the programmer has read data from the data bus, it generates a low-to-high transition on the HSP line, and as a response, the LPC1/2/4000 generates a low-to-high transition on the HSC line within the next 5 microseconds.

Sequence described above must be applied for every byte that has to be read by the programmer. Reading data from the LPC1/2/4000 device does not depend on the current Status code.

DESCRIPTION OF COMMUNICATION PROCEDURES BETWEEN THE PARALLEL PROGRAMMER AND THE PROGRAMMED LPC1/2/4000 DEVICE

This chapter describes communication procedures that can be initiated by the programmer. Every procedure starts with a command cycle (see Figure 83). Depending on the character of the procedure, several command and read cycles (see Figure 84) may follow.

For every command/read cycle in a given procedure, a list of exchanged data on the data bus is presented. The first byte exchanged in a specific cycle is marked as the byte number one. Status codes are listed in Table 59.

Procedure: Read PartID

This procedure reads the 32 bit PartID. It requires two command cycles and one read cycle. LPC18xx and LPC43xx devices provide a two 32-bit wide PartID.

Table 36: Read PartID Procedure

Cycle	Description	Byte Number	LPC11/12/13/17xx LPC2000 LPC40xx Data (D[7:0]) Hex	LPC18xx LPC43xx Data(D[7:0]) Hex	Delay CMD - RES (Status update)
Command1	Command for reading PartID	1	0x36		<20 microseconds
		2	0x00		
		3	0x00		
		4	0x00		
Command2	Command for the LPC1/2/4000 to get ready to output data to the data bus	1	0x00		< 5 microseconds
		2	0x02		
		3	0x00		
		4	0x00		
Read1	The LPC1/2/4000 outputs its own PartID on the data bus	1	0xXX		Status and RES lines do not change
		2	0xXX		
		3	0xXX		
		4	0xXX		
		5	PartID[07:00]	PartID_word0[07:00]	
		6	PartID[15:08]	PartID_word0[15:08]	
		7	PartID[23:16]	PartID_word0[23:16]	
		8	PartID[31:24]	PartID_word0[31:24]	
		9	Not Available	PartID_word1[07:00]	
		10		PartID_word1[15:08]	
		11		PartID_word1[23:16]	
		12		PartID_word1[31:24]	

Status lines should read CMD_SUCCESS (code 0) both after Command1 and Command2 cycles. PartIDs for supported LPC1/2/4000 devices are listed in Table 37

Table 37: LPC1000/2000/4000 PartIDs

Part Name	Pincount	RAM (kB)	Flash(kB)	PartID[31:00]
LPC1111/002	20	2	8	0x0A16 D02B 0x1A16 D02B
LPC1111/101	33	2	8	0x041E 502B 0x2516 D02B
LPC1111/102	33	2	8	0x2516 D02B
LPC1111/103	33	2	8	0x0001 0013
LPC1111/201	33	4	8	0x0416 502B 0x2516 902B
LPC1111/202	33	4	8	0x2516 902B

Table 37: LPC1000/2000/4000 PartIDs

Part Name	Pincount	RAM (kB)	Flash(kB)	PartID[31:00]
LPC1111/203	33	4	8	0x0001 0012
LPC11A02	20	4	16	0x4D4C 802B
LPC11A04	20	8	32	0x4D80 002B
LPC11A11/001	33	2	8	0x455E C02B
LPC11E11/101	33	4	8	0x293E 902B
LPC1112/101	33	2	16	0x042D 502B 0x2524 D02B
LPC1112/102	20/28	4	16	0x0A24 902B 0x1A24 902B
LPC1112/102	33	2	16	0x2524 D02B
LPC1112/103	33	2	16	0x0002 0023
LPC1112/201	33	4	16	0x0425 502B 0x2524 902B
LPC1112/202	24	4	16	0x2524 902B
LPC1112/202	33	4	16	0x2524 902B
LPC1112/203	33	4	16	0x0002 0022
LPC11A12/101	33/48	4	16	0x4574 802B
LPC11C12/301	48	8	16	0x1421 102B
LPC11E12/201	48	6	16	0x2954 502B
LPC11U12/201	33/48	6	16	0x095C 802B 0x295C 802B
LPC1113/201	33	4	24	0x0434 502B 0x2532 902B
LPC1113/202	33	4	24	0x2532 902B
LPC1113/203	33	4	24	0x0003 0032
LPC1113/301	33	8	24	0x0434 102B 0x2532 102B
LPC1113/302	33	8	24	0x2532 102B
LPC1113/303	33	8	24	0x0003 0030
LPC1113/301	48	8	24	0x0434 102B 0x2532 102B
LPC1113/302	48	8	24	0x2532 102B
LPC1113/303	48	8	24	0x0003 0030
LPC11A13/201	33	6	24	0x458A 402B
LPC11E13/301	48	8	24	0x296A 102B
LPC11U13/201	48	6	24	0x097A 802B 0x297A 802B
LPC1114/102	28	4	32	0x0A40 902B 0x1A40 902B
LPC1114/201	33	4	32	0x0444 502B 0x2540 902B
LPC1114/202	33	4	32	0x2540 902B
LPC1114/203	33	4	32	0x0004 0042

Table 37: LPC1000/2000/4000 PartIDs

Part Name	Pincount	RAM (kB)	Flash(kB)	PartID[31:00]
LPC1114/301	33	8	32	0x0444 102B 0x2540 102B
LPC1114/302	33	8	32	0x2540 102B
LPC1114/303	33	8	32	0x0004 0040
LPC1114/333	33	8	56	0x0004 0070
LPC1114/301	44	8	32	0x0444 102B 0x2540 102B
LPC1114/302	44	8	32	0x2540 102B
LPC1114/301	48	8	32	0x0444 102B 0x2540 102B
LPC1114/302	48	8	32	0x2540 102B
LPC1114/303	48	8	32	0x0004 0040
LPC1114/323	48	8	48	0x0004 0060
LPC1114/333	48	8	56	0x0004 0070
LPC1101LV	25	2	32	0x2714 302B
LPC1102LV	25	8	32	0x2724 002B
LPC1112LV/003	24	2	16	0x2744 202B
LPC1114LV/103	24	4	32	0x2744 202B
LPC1112LV/103	33	4	16	0x2722 202B
LPC1114LV/303	24/33	8	32	0x2744 002B
LPC11A14/301	33/48	8	32	0x35A0 002B 0x45A0 002B
LPC11C14/301	48	8	32	0x1440 102B
LPC11D14/302	100	8	32	0x2540 102B
LPC11E14/401	33	10	32	0x2980 102B
LPC11E14/401	48	10	32	0x2980 102B
LPC11E14/401	64	10	32	0x2980 102B
LPC11U14/201	33/48	6	32	0x0998 802B 0x1998 802B 0x2998 802B
LPC1115/303	48	8	64	0x0005 0080
LPC11C22/301	48	8	16	0x1431 102B
LPC11C24/301	48	8	32	0x1430 102B
LPC11U23/301	48	8	24	0x2972 402B
LPC11U24/301	33	8	32	0x2988 402B
LPC11U24/301	48	8	32	0x2988 402B
LPC11U24/401	33	10	32	0x2980 002B
LPC11U24/401	48	10	32	0x2980 002B
LPC11U24/401	64	10	32	0x2980 002B
LPC11U34/311	33/48	8	40	0x0003 D440
LPC11U34/421	33/48	10	48	0x0001 CC40
LPC11U35/401	33/48	10	64	0x0001 BC40

Table 37: LPC1000/2000/4000 PartIDs

Part Name	Pincount	RAM (kB)	Flash(kB)	PartID[31:00]
LPC11U35/401	64	10	64	0x0000 BC40 0x0001 BC40
LPC11U35/501	33/48	12	64	0x0000 BC40
LPC11U36/401	48/64	10	96	0x0001 9C40
LPC11U37/401	48/64	10	128	0x0000 7C40 0x0001 7C40
LPC11U37/501	64	12	128	0x0000 7C40
LPC1224/101	48/64	4	32	0x3640 C02B
LPC1224/121	48/64	4	48	0x3642 C02B
LPC1225/301	48/64	8	64	0x3650 002B
LPC1225/321	48/64	8	80	0x3652 002B
LPC1226/301	48/64	8	96	0x3660 002B
LPC1227/301	48/64	8	128	0x3670 002B
LPC12D27/301	100	8	128	0x3670 002B
LPC1311	33	4	8	0x2C42 502B
LPC1311/01	33	4	8	0x1816 902B
LPC1313	33/48	8	32	0x2C40 102B
LPC1313/01	33/48	8	32	0x1830 102B
LPC1315	33/48	8	32	0x3A01 0523
LPC1316	33/48	8	48	0x1A01 8524
LPC1317	33/48/64	10	64	0x1A02 0525
LPC1342	33	4	16	0x3D01 402B
LPC1343	33/48	8	32	0x3D00 002B
LPC1345	33/48	10	32	0x2801 0541
LPC1346	33/48	10	48	0x0801 8542
LPC1347	33/48/64	12	64	0x0802 0543
LPC1751	80	8	32	0x2500 1110
LPC1752	80	16	64	0x2500 1121
LPC1754	80	32	128	0x2501 1722
LPC1756	80	32	256	0x2501 1723
LPC1758	80	64	512	0x2501 3F37
LPC1759	80	64	512	0x2511 3737
LPC1763	100	64	256	0x2601 2033
LPC1764	100	32	128	0x2601 1922
LPC1765	100	64	256	0x2601 3733
LPC1766	100	64	256	0x2601 3F33
LPC1767	100	64	512	0x2601 2837
LPC1768	100	64	512	0x2601 3F37
LPC1769	100	64	512	0x2611 3F37
LPC1774	144/208	40	128	0x2701 1132
LPC1776	180/208	80	256	0x2719 1F43
LPC1777	208	96	512	0x2719 3747
LPC1778	144/180/208	96	512	0x2719 3F47

Table 37: LPC1000/2000/4000 PartIDs

Part Name	Pincount	RAM (kB)	Flash(kB)	PartID[31:00]
LPC1785	208	80	256	0x281D 1743
LPC1786	208	80	256	0x281D 1F43
LPC1787	208	96	512	0x281D 3747
LPC1788	144/180/208	96	512	0x281D 3F47
LPC1812	100/144	104	512	word0: 0xF00B DB3F word1: 0x0000 0080
LPC1813	100/144	104	512	word0: 0xF00B DB3F word1: 0x0000 0044
LPC1815	100/144	136	768	word0: 0xF001 DB3F word1: 0x0000 0022
LPC1817	100/144	136	1024	word0: 0xF001 DB3F word1: 0x0000 0000
LPC1822	100/144	104	512	word0: 0xF00B DB3C word1: 0x0000 0080
LPC1823	100/144	104	512	word0: 0xF00B DB3C word1: 0x0000 0044
LPC1825	100/144	136	768	word0: 0xF001 DB3C word1: 0x0000 0022
LPC1827	100/144	136	1024	word0: 0xF001 DB3C word1: 0x0000 0000
LPC1833	100/144/256	136	512	word0: 0xF001 DA30 word1: 0x000 00044
LPC1837	100/144/256	136	1024	word0: 0xF001 DA30 word1: 0x0000 0000
LPC1853	208/256	136	512	word0: 0xF001 D830 word1: 0x0000 0044
LPC1857	208/256	136	1024	word0: 0xF001 D830 word1: 0x0000 0000
LPC2101	48	2	8	0x0004 FF11
LPC2102	48	4	16	0x0004 FF11
LPC2103	48	8	32	0x0004 FF11
LPC2104	48	16	128	0xFFF0 FF12
LPC2105	48	32	128	0xFFF0 FF22
LPC2106	48	64	128	0xFFF0 FF32
LPC2109	64	64	8	0x0201 FF01
LPC2114	64	16	128	0X0101 FF12
LPC2119	64	16	128	0X0201 FF12
LPC2124	64	16	256	0X0101 FF13
LPC2129	64	16	256	0X0201 FF13
LPC2194	64	16	256	0X0301 FF13
LPC2131	64	8	32	0X0002 FF01
LPC2132	64	16	64	0X0002 FF11
LPC2134	64	16	128	0X0002 FF12
LPC2136	64	32	256	0X0002 FF23

Table 37: LPC1000/2000/4000 PartIDs

Part Name	Pincount	RAM (kB)	Flash(kB)	PartID[31:00]
LPC2138	64	32	512	0X0002 FF25
LPC2141	64	8	32	0x0402 FF01
LPC2142	64	16	64	0X0402 FF11
LPC2144	64	16	128	0X0402 FF12
LPC2146	64	32	256	0X0402 FF23
LPC2148	64	32	512	0X0402 FF25
LPC2212	144	16	128	0X0401 FF12
LPC2214	144	16	256	0X0601 FF13
LPC2292	144	16	256	0X0401 FF13
LPC2294	144	16	256	0X0501 FF13
LPC2361	100	34	64	0x1600 F701
LPC2362	100	58	128	0x1600 FF22
LPC2364	100	34	128	0x0603 FB02 0x1600 F902
LPC2365	100	58	256	0x1600 E823
LPC2366	100	58	256	0x0603 FB23 0x1600 E123 0x1600 F923
LPC2367	100	58	512	0x1600 E825
LPC2368	100	58	512	0x0603 FB25 0x1600 F925
LPC2377	144	58	512	0x1700 E825
LPC2378	144	58	512	0x0703 FF25 0x1700 FD25
LPC2387	100	98	512	0x1700 FF35 0x1800 F935
LPC2388	144	98	512	0x1800 FF35
LPC2458	180	98	512	0x1500 E735 0x1500 FF35
LPC2468	208	98	512	0x0603 FF35 0x1600 FF35
LPC2478	208	98	512	0x1701 FF35
LPC4072	80	16	64	0x4701 1121
LPC4074	80/144	40	128	0x4701 1132
LPC4076	144/180	80	256	0x4719 1F43
LPC4078	80/100/144/ 180/208	96	512	0x4719 3F47
LPC4088	144/180/208	96	512	0x481D 3F47
LPC4312	100/144	104	512	word0: 0xA00B CB3F word1: 0x0000 0080
LPC4313	100/144	104	512	word0: 0xA00B CB3F word1: 0x0000 0044
LPC4315	100/144	136	768	word0: 0xA001 CB3F word1: 0x0000 0022

Table 37: LPC1000/2000/4000 PartIDs

Part Name	Pincount	RAM (kB)	Flash(kB)	PartID[31:00]
LPC4317	100/144	136	1024	word0: 0xA001 CB3F word1: 0x0000 0000
LPC4322	100/144	104	512	word0: 0xA00B CB3C word1: 0x0000 0080
LPC4323	100/144	104	512	word0: 0xA00B CB3C word1: 0x0000 0044
LPC4325	100/144	136	768	word0: 0xA001 CB3C word1: 0x0000 0022
LPC4327	100/144	136	1024	word0: 0xA001 CB3C word1: 0x0000 0000
LPC4333	100/144/256	136	512	word0: 0xA001 CA30 word1: 0x0000 0044
LPC4337	100/144/256	136	1024	word0: 0xA001 CA30 word1: 0x0000 0000
LPC4353	208/256	136	512	word0: 0xA001 C830 word1: 0x0000 0044
LPC4357	208/256	136	1024	word0: 0xA001 C830 word1: 0x0000 0000

Important details for handling LPC2101, LPC2102, and LPC2103 devices

LPC2101, LPC2102, and LPC2103 share the same PartID. Therefore special attention is needed when these devices are programmed.

When erasing the LPC2101/2102/2103 Flash: if the PartID retrieved from the chip has a value of 0x0004 FF11, the full chip erase operation has to be carried on as if it were a 32 kB Flash LPC2103 device (i.e. by specifying sectors from 0 to 7).

When programming the LPC2101/2102/2103 Flash: the parallel programmer should act according to what the user has selected. For example, the customer may select a LPC2101 to be programmed. Even though the PartID read from the chip will say 0x0004 FF11, the parallel programmer must warn this customer if he/she attempts to load and/or program any content that exceeds 8 kB of Flash.

Procedure: Read Boot Code Version

This procedure reads the Boot Code Version as an 8 byte data. They are organized as CodeVersionI (4 bytes) and CodeVersionP (4 bytes). The CodeVersionI is the Boot Code Version from the ISP software that handles initial serial communication, while the CodeVersionP refers to the code written for two-way/parallel handshake/data transfer. This procedure requires two command cycles and one read cycle.

Table 38: Read Boot Code Version Procedure

Cycle	Description	Byte Number	Data (D[7:0]) Hex	Delay CMD - RES (Status update)
Command1	Command for reading Boot Code Version	1	0x37	<20 microseconds
		2	0x00	
		3	0x00	
		4	0x00	
Command2	Command for the LPC1/2/4000 to get ready to output data to the data bus	1	0x00	< 5 microseconds
		2	0x02	
		3	0x00	
		4	0x00	
Read1	The LPC1/2/4000 outputs Boot Code Version on the data bus	1	0xXX	Status and RES lines do not change
		2	0xXX	
		3	0xXX	
		4	0xXX	
		5	CodeVersionI.[0]	
		6	CodeVersionI.[1]	
		7	CodeVersionI.[2]	
		8	CodeVersionI.[3]	
		9	CodeVersionP.[0]	
		10	CodeVersionP.[1]	
		11	CodeVersionP.[2]	
		12	CodeVersionP.[3]	

Status lines should read CMD_SUCCESS (code 0) both after Command1 and Command2 cycles.

Data read from the LPC2000 parts identify CodeVersionI as 1.5 and higher (coded as 0x0000 0015) and CodeVersionP as 2.0 (coded as 0x0000 0020) and higher.

Procedure: Read Memory

This procedure reads content from a single or multiple memory locations starting with an address location MemAddr[31:00]. This command can read both from the RAM and Flash memories. Number of bytes to be read is determined by the number of performed read cycles. This procedure requires one command cycle and a number of read cycles as desired.

Important: Reading of the memory content from LPC2101/2102/2103 should be performed as per memory layout outlined in Table 37. Only 8 kB of Flash should be read from a LPC2101 and only 16 kB of Flash should be read from a LPC2102. In case of a LPC2103 device full 32 kB of Flash can be read.

Table 39: Read Memory Procedure

Cycle	Description	Byte Number	Data (D[7:0]) Hex	Delay CMD - RES (Status update)
Command1	Command for the LPC1/2/4000 to get ready to output data to the data bus beginning with memory location MemAddr[31:00]. **Important:** MemAddr parameter should be within the range of addresses specified in Table 17, Table 18 or Table 20. Readings out of these ranges are subject to errors.	1 2 3 4 5 6 7 8	0x00 0x03 0x00 0x00 MemAddr[07:00] MemAddr[15:08] MemAddr[23:16] MemAddr[31:24]	< 5 microseconds
Read1	The LPC1/2/4000 outputs data from location MemAddr[31:00]+0	1	(MemAddr[31:00]+0)	Status and RES lines do not change
Read2	The LPC1/2/4000 outputs data from location MemAddr[31:00]+1	1	(MemAddr[31:00]+1)	Status and RES lines do not change
Read3	The LPC1/2/4000 outputs data from location MemAddr[31:00]+2	1	(MemAddr[31:00]+2)	Status and RES lines do not change
⋮	⋮	⋮	⋮	⋮

Status lines should read CMD_SUCCESS (code 0) after Command1 cycle.

MemAddr parameter supplied by the parallel programmer in Command1 points to the beginning of the memory block that will be read in following Read cycles. After a single read cycle is completed, the LPC1/2/4000 will auto increment memory address, read its content and wait for a new Read cycle to be initiated by the parallel programmer.

Procedure: Erase Flash Memory

This procedure erases a single or multiple Flash memory sectors of the on-chip Flash memory starting with the sector StartSectNo and ending with the EndSectNo, including these two sectors, too. Specifying sector 0 as the StartSecNo and the last available Flash sector on the programmed LPC1/2/4000 device (see Table 17, Table 18, Table 20, and Table 22 for details) will perform full chip erase of the microcontroller. This procedure requires two command cycles.

LPC18xx and LPC43xx microcontrollers require an additional parameter, the Flash bank number - FlashBankNo, to be specified. For details on the Flash banks and sectors address ranges see Table 19.

Important: LPC2101/2102 full chip erase is handled as in case of LPC2103 by selecting sectors from 0 to 7.

Table 40: Erase Flash Memory Procedure (LPC2xxx)

Cycle	Description	Byte #	Data (D[7:0]) Hex		Delay CMD - RES (Status update)
			LPC21xx/ 22xx	LPC23xx/ 24xx	
Command1	Command for the LPC2000 to prepare sectors from StartSectNo to EndSectNo for Erase. **Important:** StartSectNo and EndSectNo can have any value from 0 to the last available Flash sector number of the used microcontroller. StartSectNo must be less than or equal EndSectNo. If StartSectNo equals EndSectNo, only this single sector will be prepared for Erase.	1	0x32		< 200 microseconds
		2	0x00		
		3	0x00		
		4	0x00		
		5	StartSectNo		
		6	0x00		
		7	0x00		
		8	0x00		
		9	EndSectNo		
		10	0x00		
		11	0x00		
		12	0x00		
Command2	Command for LPC2000 to erase Flash memory sectors from StartSectNo to EndSectNo. **Important:** StartSectNo and EndSectNo in Command2 must match StartSectNo and EndSectNo from the above Command1.	1	0x34		Erase of any number of Flash memory Sectors lasts fixed 400 milliseconds. High-to-low transition should not appear on RES line before 400 milliseconds expire, nor later then 401 milliseconds after a low-to-high transition occurred on the CMD line.
		2	0x00		
		3	0x00		
		4	0x00		
		5	StartSectNo		
		6	0x00		
		7	0x00		
		8	0x00		
		9	EndSectNo		
		10	0x00		
		11	0x00		
		12	0x00		
		13	0x40	0x40	
		14	0x9C	0x19	
		15	0x00	0x01	
		16	0x00	0x00	

Table 41: Erase Flash Memory Procedure (LPC11/12/13/17xx/40xx)

Cycle	Description	Byte #	Data (D[7:0]) Hex																Delay CMD - RES (Status update)
			LPC111x LPC11xLV LPC11Cxx LPC11Dxx LPC11Exx LPC11Uxx LPC1315							24-pin LPC1112, LPC11Axx			LPC122x LPC12Dx		LPC1311 LPC175x LPC176x		LPC177x LPC178x LPC40xx		
Command1	Command for the LPC11/12/13/17xx/40xx to prepare sectors from StartSectNo to EndSectNo for Erase. **Important:** StartSectNo and EndSectNo can have any value from 0 to the last available Flash sector number of the used microcontroller. StartSectNo must be less than or equal EndSectNo. If StartSectNo equals EndSectNo, only this single sector will be prepared for Erase.	1	0x32																< 200 microseconds
		2	0x00																
		3	0x00																
		4	0x00																
		5	StartSectNo																
		6	0x00																
		7	0x00																
		8	0x00																
		9	EndSectNo																
		10	0x00																
		11	0x00																
		12	0x00																
Command2	Command for LPC11/12/13/17xx/40xx to erase Flash memory sectors from StartSectNo to EndSectNo. **Important:** StartSectNo and EndSectNo in Command2 must match StartSectNo and EndSectNo from the above Command1.	1	0x34																Erase of any number of Flash memory Sectors lasts fixed 100 milliseconds. High-to-low transition should not appear on RES line before 100 milliseconds expire, nor later then 101 milliseconds after a low-to-high transition occurred on the CMD line.
		2	0x00																
		3	0x00																
		4	0x00																
		5	StartSectNo																
		6	0x00																
		7	0x00																
		8	0x00																
		9	EndSectNo																
		10	0x00																
		11	0x00																
		12	0x00																
		13	0x50							0x80		0x30		0x70		0xA0			
		14	0xC3							0xBB		0x75		0x11		0x86			
		15	0x00							0x00		0x00		0x01		0x01			
		16	0x00							0x00		0x00		0x00		0x00			

Important: In case of the LPC18xx and LPC43xx microcontrollers, the Erase Flash Memory procedure can access sectors in the selected Flash bank only. To perform full chip erase it is necessary to erase all sectors in the Flash bank 0 independently from the sectors in the Flash bank 1.

.

Table 42: Erase Flash Memory Procedure (LPC18xx and LPC43xx)

Cycle	Description	Byte #	Data (D[7:0]) Hex	Delay CMD - RES (Status update)
Command1	Command for the LPC18xx and LPC43xx to prepare sectors from StartSectNo to EndSectNo in the selected FlashBankNo for Erase. **Important:** StartSectNo and EndSectNo can have any value from 0 to the last available Flash sector number of the used microcontroller's Flash Bank. StartSectNo must be less than or equal EndSectNo. If StartSectNo equals EndSectNo, only this single sector will be prepared for Erase.	1	0x32	< 200 microseconds
		2	0x00	
		3	0x00	
		4	0x00	
		5	StartSectNo	
		6	0x00	
		7	0x00	
		8	0x00	
		9	EndSectNo	
		10	0x00	
		11	0x00	
		12	0x00	
		13	FlashBankNo	
		14	0x00	
		15	0x00	
		16	0x00	
Command2	Command for LPC18xx and LPC43xx to erase Flash memory sectors from StartSectNo to EndSectNo in the selected FlashBankNo. **Important:** StartSectNo and EndSectNo in Command2 must match StartSectNo, EndSectNo and FlashBankNo from the above Command1.	1	0x34	Erase of any number of Flash memory Sectors lasts fixed 400 milliseconds. High-to-low transition should not appear on RES line before 400 milliseconds expire, nor later then 401 milliseconds after a low-to-high transition occurred on the CMD line.
		2	0x00	
		3	0x00	
		4	0x00	
		5	StartSectNo	
		6	0x00	
		7	0x00	
		8	0x00	
		9	EndSectNo	
		10	0x00	
		11	0x00	
		12	0x00	
		13	0x00	
		14	0x77	
		15	0x01	
		16	0x00	
		17	FlashBankNo	
		18	0x00	
		19	0x00	
		20	0x00	

The Status lines should read CMD_SUCCESS (code 0) after Command1/Command2 cycles. If either StartSectNo

or EndSectNo is out of the range of available Flash sectors, an INVALID_SECTOR (code 0x07) will appear on the Status lines. If StartSectNo and/or EndSectNo in Command2 do not match selection in Command1, a SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION (code 0x09) may appear on Status lines.

Procedure: Blank Check

This procedure blank checks a single or multiple sectors of the LPC1/2/4000 on-chip Flash memory. It requires one command cycle only. In case a non-blank word location is found, its location and content will be available for reading. In this case, an additional command and read cycles are required.

LPC18xx and LPC43xx microcontrollers require an additional parameter, the Flash bank number - FlashBankNo, to be specified. For details on the Flash banks and sectors address ranges see Table 19.

Important: when performing a blank chack on LPC2101/2102/2103 it has to be handled identically to the erase operation. This means that for a full chip blank check StartSectNo=0 and EndSectNo=7.

Table 43: Blank Check Procedure

Cycle	Description	Byte #	Data (D[7:0]) Hex		Delay CMD - RES (Status update)
			LPC11/12/13/15/17xx LPC2000 LPC40xx	LPC18xx LPC43xx	
Command1	Command for LPC1/2/4000 to perform blank check on sectors from StartSectNo to EndSectNo for Erase. **Important:** StartSectNo and EndSectNo can have any value from 0 to the last available Flash sector number of the used microcontroller. StartSectNo must be less than or equal EndSectNo. If StartSectNo equals EndSectNo, only this single sector will be blank checked. LPC18xx and LPC43xx devices must specify the Flash bank that is accessed.	1	0x35		< 100 milliseconds
		2	0x00		
		3	0x00		
		4	0x00		
		5	StartSectNo		
		6	0x00		
		7	0x00		
		8	0x00		
		9	EndSectNo		
		10	0x00		
		11	0x00		
		12	0x00		
		13	Not Applicable	FlashBankNo	
		14		0x00	
		15		0x00	
		16		0x00	
Sequence for obtaining more data in case Status lines code reads 0x08 (SECTOR_NOT_BLANK) after the Command1 cycle is finished:					
Command2	Command sent to the LPC1/2/4000 to prepare for output of the data to the data bus	1	0x00		< 5 microseconds
		2	0x02		
		3	0x00		
		4	0x00		

Table 43: Blank Check Procedure

Cycle	Description	Byte #	Data (D[7:0]) Hex		Delay CMD - RES (Status update)
			LPC11/12/13/15/17xx LPC2000 LPC40xx	LPC18xx LPC43xx	
Read1	LPC1/2/4000 outputs SECTOR_NOT_BLANK code, an 32-bit address (NonBlankAddr[31:00]) of non-blank word, as well as the very non-blank 32-bit content (NonBlankWord[31:00]).	1	0x08		Status and RES lines do not change
		2	0x00		
		3	0x00		
		4	0x00		
		5	NonBlankAddr[07:00]		
		6	NonBlankAddr[15:08]		
		7	NonBlankAddr[23:16]		
		8	NonBlankAddr[31:24]		
		9	NonBlankWord[07:00]		
		10	NonBlankWord[15:08]		
		11	NonBlankWord[23:16]		
		12	NonBlankWord[31:24]		

Status lines should read CMD_SUCCESS (code 0x00) after Command1 cycle.

If either StartSectNo or EndSectNo is out of range of the selected LPC1/2/4000 available Flash sectors, an INVALID_SECTOR (code 0x07) will appear on the Status lines.

If there is a single non-blank word in the Flash memory within [StartSectNo:EndSectNo sectors] range, SECTOR_NOT_BLANK (code 0x08) will appear on the Status lines. The procedure for identifying the first non-blank location and its content is shown in the table above.

Procedure: Program Flash Memory and Verify

This procedure loads 1 kB of data into the microcontroller's RAM, programs these data into the Flash sector FlashSect starting as of address FlashAddr, and verifies that the corresponding RAM and Flash content match. It requires three command cycles.

LPC18xx and LPC43xx microcontrollers require an additional parameter, the Flash bank number - FlashBankNo, to be specified. For details on the Flash banks and sectors address ranges see Table 19.

Details on selecting FlashSect and FlashAddr follow.

Table 44: LPC11/12/13/17xx/2000/40xx Program Flash Memory and Verify Procedure

Cycle	Description	Byte Number	Data (D[7:0]) Hex							Delay CMD - RES (Status update)
			LPC21xx/22xx	LPC23xx/24xx	LPC111x, LPC11xxLV LPC11Cxx, LPC11Dxx, LPC11Exx, LPC11Uxx LPC1315/16/17/45/46/47	24-pin LPC1112, LPC11Axx	LPC122x LPC12Dxx	LPC1311/13/42/43 LPC175x/6x	LPC177x/8x LPC407x/8x	
Command1	Command for LPC1000/2000/40xx to prepare Flash sector FlashSect for programming.	1	0x32							< 200 microseconds
		2	0x00							
		3	0x00							
		4	0x00							
		5	FlashSect							
		6	0x00							
		7	0x00							
		8	0x00							
		9	FlashSect							
		10	0x00							
		11	0x00							
		12	0x00							

Table 44: LPC11/12/13/17xx/2000/40xx Program Flash Memory and Verify Procedure

Cycle	Description	Byte Number	Data (D[7:0]) Hex							Delay CMD - RES (Status update)
			LPC21xx/22xx	LPC23xx/24xx	LPC111x, LPC11xxLV LPC11Cxx, LPC11Dxx, LPC11Exx, LPC11Uxx LPC1315/16/17/45/46/47	24-pin LPC1112, LPC11Axx	LPC122x LPC12Dxx	LPC1311/13/42/43 LPC175x/6x	LPC177x/8x LPC407x/8x	
Command2	Command for LPC1000/2000/40xx to load data into the RAM and copy them to the Flash.	1	0x33							Time required for data to be loaded from the programer to the device depends on the capabilities of the programming equipment. For details see Figure 83. After 1kB of data is loaded, it takes less than 10 milliseconds for them to be programmed.
		2	0x00							
		3	0x00							
		4	0x00							
		5	FlashAddr[07:00]							
		6	FlashAddr[15:08]							
		7	FlashAddr[23:16]							
		8	FlashAddr[31:24]							
		9	0x14	0x14	0x14	0x14	0x14	0x14		
		10	0x05	0x07	0x00	0x00	0x00	0x0A		
		11	0x00	0x00	0x00	0x00	0x00	0x00		
		12	0x40	0x40	0x10	0x10	0x10	0x10		
		13	0x00							
		14	0x04							
		15	0x00							
		16	0x00							
		17	0x40	0x40	0x50	0x80	0x30	0x70	0xA0	
		18	0x9C	0x19	0xC3	0xBB	0x75	0x11	0x86	
		19	0x00	0x01	0x00	0x00	0x00	0x01	0x01	
		20	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
Command2 (continued)	data byte 0 out of 1 kB	21	$byte_0$[7:0]							
	data byte 1 out of 1 kB	22	$byte_1$[7:0]							
	⋮	⋮	⋮							
	data byte 1023 out of 1 kB	20+ 1024	$byte_{1023}$[7:0]							

Table 44: LPC11/12/13/17xx/2000/40xx Program Flash Memory and Verify Procedure

Cycle	Description	Byte Number	Data (D[7:0]) Hex							Delay CMD - RES (Status update)
			LPC21xx/22xx	LPC23xx/24xx	LPC111x, LPC11xxLV LPC11Cxx, LPC11Dxx, LPC11Exx, LPC11Uxx LPC1315/16/17/45/46/47	24-pin LPC1112, LPC11Axx	LPC122x LPC12Dxx	LPC1311/13/42/43 LPC175x/6x	LPC177x/8x LPC407x/8x	
Command3	Command for LPC1000/2000/40xx to compare data from the RAM and those programmed into the Flash.	1	0x38							< 2 milliseconds
		2	0x00							
		3	0x00							
		4	0x00							
		5	FlashAddr[07:00]							
		6	FlashAddr[15:08]							
		7	FlashAddr[23:16]							
		8	FlashAddr[31:24]							
		9	0x14	0x14	0x14	0x14	0x14	0x14		
		10	0x05	0x07	0x00	0x00	0x00	0x0A		
		11	0x00	0x00	0x00	0x00	0x00	0x00		
		12	0x40	0x40	0x10	0x10	0x10	0x10		
		13	0x00							
		14	0x04							
		15	0x00							
		16	0x00							

Details on selecting FlashSect and FlashAddr follow.

Table 45: LPC18xx and LPC43xx Program Flash Memory and Verify Procedure

Cycle	Description	Byte Number	Data (D[7:0]) Hex	Delay CMD - RES (Status update)
Command1	Command for LPC18xx and LPC43xx to prepare Flash sector FlashSect in the selected Flash bank FlashBankNo for programming.	1	0x32	< 200 microseconds
		2	0x00	
		3	0x00	
		4	0x00	
		5	FlashSect	
		6	0x00	
		7	0x00	
		8	0x00	
		9	FlashSect	
		10	0x00	
		11	0x00	
		12	0x00	
		13	FlashBankNo	
		14	0x00	
		15	0x00	
		16	0x00	
Command2	Command for LPC18xx and LPC43xx to load data into the RAM and copy them to the Flash.	1	0x33	Time required for data to be loaded from the programer to the device depends on the capabilities of the programming equipment. For details see Figure 83. After 1kB of data is loaded, it takes less than 10 milliseconds for them to be programmed.
		2	0x00	
		3	0x00	
		4	0x00	
		5	FlashAddr[07:00]	
		6	FlashAddr[15:08]	
		7	FlashAddr[23:16]	
		8	FlashAddr[31:24]	
		9	0x14	
		10	0x00	
		11	0x00	
		12	0x10	
		13	0x00	
		14	0x04	
		15	0x00	
		16	0x00	
		17	0x00	
		18	0x77	
		19	0x01	
		20	0x00	

Table 45: LPC18xx and LPC43xx Program Flash Memory and Verify Procedure

Cycle	Description	Byte Number	Data (D[7:0]) Hex	Delay CMD - RES (Status update)
Command2 (continued)	data byte 0 out of 1 kB	21	$byte_0$[7:0]	
	data byte 1 out of 1 kB	22	$byte_1$[7:0]	
	⋮	⋮	⋮	
	data byte 1023 out of 1 kB	20+ 1024	$byte_{1023}$[7:0]	
Command3	Command for LPC18xx and LPC43xx to compare data from the RAM and those programmed into the Flash.	1	0x38	< 2 milliseconds
		2	0x00	
		3	0x00	
		4	0x00	
		5	FlashAddr[07:00]	
		6	FlashAddr[15:08]	
		7	FlashAddr[23:16]	
		8	FlashAddr[31:24]	
		9	0x14	
		10	0x00	
		11	0x00	
		12	0x10	
		13	0x00	
		14	0x04	
		15	0x00	
		16	0x00	

Status lines should read CMD_SUCCESS (code 0) after Command1, Command2 and Command3 cycles.

Command1 considerations:

If FlashSect is out of the selected LPC1/2/4000 Flash sector's range, an INVALID_SECTOR (code 0x07) will show up on the Status lines.

Command2 considerations:

FlashAddr must be a 512 byte boundary, i.e. FlashAddr must be a multiple of 512. Otherwise, an DST_ADDR_ERROR (code 0x03) will show up after the Command2.

It is important that both FlashAddr and FlashAddr+1023 reside in the same sector selected by the FlashSect in Command1. In case either FlashAddr or FlashAddr+1023 is out of the FlashSect sector boundaries (Table 17), a SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION (code 0x09) will show up on the Status lines.

Command3 considerations:

FlashAddr must be the same as in the Command2.

Flash Memory Code Read protection

LPC1/2/4000 Flash memory Code Read Protection (CRP) mechanism combines microcontroller's hardware and factory built in on-chip software to prevent an unauthorised reading of the on-chip Flash memory. This feature is not available on LPC2104/5/6 devices. Also, the CRP is not supported by older reversions of the on-chip bootloader on some of the LPC2000 parts. Here is the list of devices and corresponding on-chip bootloaders that fully support the CRP:

Table 46: LPC1000/2000/4000 devices and on-chip bootloader versions that support the CRP

Device name	On-chip bootloader rev
LPC1111/1112/1113/1114	7.0
LPC1101LV/1102LV/1112LV/1114LV	7.0
LPC11A02/11A04/11A11/11A12/11A13/11A14	7.2
LPC11C12/11C14	7.1
LPC11C22/11C24	7.0
LPC11D14	7.1
LPC11E11/11E12/11E13/11E14	7.0
LPC11U12/11U13/11U14/11U23/11U24/11U34/11U35/11U36/11U37	7.0
LPC1224/1225/1226/1227	6.1
LPC12D27	6.1
LPC1311/1313/1342/1343	5.0
LPC1315/1316/1317/1345/1346/1347	5.2
LPC1751/1752/1754/1756/1758/1759	4.1
LPC1763/1764/1765/1766/1767/1768/1769	4.1
LPC1774/1776/1777/1778	8.1
LPC1785/1786/1787/1788	8.1
LPC1812/1813/1815/1817/1822/1823/1825/1827/1833/1837/1853/1857	3072.1
LPC2101/2102/2103	2.2
LPC2109/2112/2119/2124/2129/2194	1.6
LPC2131/2132/2134/2136/2138	2.0
LPC2212/2214/2292/2294	1.6
LPC2364/2365/2366/2367/2368	3.0
LPC2377/2378	3.0
LPC2387/2388	3.0
LPC2458/2468/2478	3.0
LPC4072/4074/4076/4078/4088	8.1
LPC4312/4313/4315/4317/4322/4323/4325/4327/4333/4337/4353/4357	3072.1

Devices not listed in Table 46 do not support the CRP. Also, the on-chip bootloader revisions (see CodeVersionI in Procedure: Read Boot Code Version on page 141) older than listed in this table will not provide support for the CRP.

IMPORTANT: The application image provided for flash programming should at a minimum have the CRP location blanked out (for example set to 0xFFFF_FFFF) or alternatively have the CRP location programmed with the desired CRP level.

LPC1000/4000 Code read protection is enabled by programming the Flash address location 0x2FCH (flash sector 0) with the desired CRP value.

LPC2000 Code read protection is enabled by programming the Flash address location 0x1FCH (flash sector 0) with the desired CRP value.

NOTE: CRP is operational only if the CRP level is residing in location 0x2FCH (LPC1000/4000) or 0x1FCH (LPC2000) when the power is applied to the device. In all other cases, CRP will be ineffective, and the microcontroller will not be protected!

The following table provides a list of the CRP levels that are available on LPC1/2/4000 devices. Depending on the device some of these CRP levels may or may not be supported. **Please consult the particular device user manual for more details on CRP and the latest information on which CRP levels are supported.**

Table 47: LPC1000/2000/4000 devices CRP level coding

CRP Level	CRP Value (HEX)
NO_ISP	0x4E69_7370
CRP1	0x1234_5678
CRP2	0x8765_4321
CRP3	0x4321_8765

Application note AN10851 available for online download from http://www.nxp.com/microcontrollers provides more details on the usage of CRP.

PROGRAMMING OF A LPC1/2/4000 DEVICE WITH THE CRP ALREADY ENABLED

This is the case when a LPC1/2/4000 device was programmed with user's code and the CRP option enabled, and now is the time to reprogramme the LPC1/2/4000 Flash with a new code.

Once the CRP mechanism is enabled and the LPC1/2/4000 device is reset, it is not possible to either write any data into the on-chip Flash memory or to read data from the microcontroller's memory. The only way the LPC1/2/4000 Flash can be programmed again is by performing a full chip erase using the ISP.

This also means that after the CRP is enabled and the device has been reset, it is not possible to either load the parallel interface code nor to use it. In case the CRP is active after a reset, the LPC1/2/4000 device will reject any kind of data download even via the ISP and UART. As a result, any attempt to run the parallel loading application from the appendix of this specification will be futile and even if all messages (from the selected device Table 24 thru Table 31) have been sent to the microcontroller, the application will not run and falling edges on the status lines (S[3:0]) and the RES line will not be generated.

The absence of confirmation that the parallel interface code is active (i.e. missing falling edges on status and RES lines) can be used to indicate active CRP on the device and the necessity to perform full chip erase before the LPC1/2/4000 chip is reprogrammed.

Full Chip Erase using the ISP

External clock signal of 10 MHz must be available to the part on pins X1/2 for the whole time of its erasing.

After power is applied to the microcontroller, reset of the device follows. It is advised that 3.3V does not come before 1.8 power supply (when 1.8V power supply is needed), since 1.8V powers the core and 3.3V enables pins to operate properly. After low level is applied to the Reset pin the ISP line must go low, too. The Reset signal has to stay low for a period of 10 milliseconds. After this, the Reset signal goes high and stays high for the rest of erasing process. The ISP line must be held low for additional 5 milliseconds after the Reset line has gone high. After this delay expires, the ISP line will be used as a data line (D6) in case one pin is used to host both the ISP and D6 function.

At this point, the ISP (In System Programming) software module residing on the LPC1/2/4000 microcontroller has been activated. The next step requires serial data load into the device.

The ISP communicates using the UART0 peripheral. Hence, the ISP serial data transfer uses UART based communication at 9600 bit/s, 1 start, 8 data, 1 stop bit format. The LPC1/2/4000 receives data on the SDIN pin. Echoed data as well as additional content is available on the SDOUT pin. While data on the SDOUT can be used for process monitoring, they are not crucial for flash erasing and subsequent programming.

Several messages have to be sent from the programmer to the LPC1/2/4000 device during the erase process. After the last byte of each message is sent over the SDIN line a specific delay is required before the next message is transmitted to the SDIN. The order of messages, their contents and the delay after each of them for full chip LPC2100/2200 device are listed in Table 48.

Handling LPC1/2/4000 devices needs minor changes (such as sector numbers, W and G commands). All relevant parameters can be found in earlier sections of this document.

Message content is case sensitive. After echoing every message on the SDOUT pin, the LPC1/2/4000 microcontroller will report if a received message is recognized as a valid one, and if so, execute it. List of error codes is available in the appendix of this document (see Table 60).

Although the 9600 bit/s baudrate is suggested in the text above, the LPC1/2/4000 is capable of establishing serial communication of 19200 and 38400 bit/s when external crystal of 10 MHz is used, too. However, this specification provides waveforms and timing related to the 9600 bit/s option only.

Table 48: Messages required to be sent over the serial channel during full chip erase (LPC21xx/22xx) and subsequent programming

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
1	'?' Note: programmer sends this character so that the microcontroller can autobaud to the 9600/19200/38400 bit/s baudrate	'Synchronized'+<CR>+<LF>	20 ms
2	'Synchronized'+<CR>+<LF>[1] Note: programmer confirms to microcontroller that the synchronization has been achieved	'Synchronized'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
3	'10000'+<CR>+<LF>[1] Note: this message informs the microcontroller that its external clock is 10 MHz	'10000'+<CR>+<LF> 'OK'+<CR>+<LF>	15 ms
4	'U 23130'+<CR>+<LF>[1] Note: this message "unlocks "the microcontroller and enables it to execute following commands	'U 23130'+<CR>+<LF> '0'+<CR>+<LF>	20 ms
5	'P 0 14'+<CR>+<LF>[1] Note: this command prepares on-chip sectors from 0 to 14 for erase operation	'P 0 14' +<CR>+<LF> '0'+<CR>+<LF>	15 ms
6	'E 0 14'+<CR>+<LF>[1] Note: this command erases on-chip sectors 0 to 14	'E 0 14'+<CR>+<LF> '0'+<CR>+<LF>	440 ms

Table 48: Messages required to be sent over the serial channel during full chip erase (LPC21xx/22xx) and subsequent programming

Message Number	Message content sent to microcontroller (data received on the SDIN)	Data microcontroller responds with/echoes on the SDOUT pin	Minimum Required Delay on the SDIN pin
7	'W 1073742336 720'+<CR>+<LF>[1] Note: this command loads 720 bytes represented as the UUENCODED data into the microcontroller's RAM starting from the address 0x4000 0200 using serial interface. Full listing of the UUENCODED data is available in the Appendix section	'W 1073742336 720' +<CR>+<LF> '0'+<CR>+<LF>	12 ms
8	'G 1073743036 A'+<CR>+<LF>[1] Note: this command executes previously loaded application staring with the instruction at address 0x4000 04BC	'G 1073743036 A'+<CR>+<LF> '0'+<CR>+<LF>	16 ms

1<CR> represents ASCII character with code 13 (0x0DH) - "Carriage Return"
<LF> represents ASCII character with code 10 (0x0AH) - "Line Feed"

Procedure of a full chip erase followed by the chip programming (Table 48) and the procedure for programming chip only (Table 24) differ in the stage of full chip erasing. The erasing itself is achieved by sending messages 5 and 6 as listed in Table 48.

Messages 7 and 8 in Table 48 related to the UUENCODED data download and the parallel interface initialization are identical to messages 5 and 6 in Table 24. Therefore all details related to them can be found in section "Initial data load Using UART0" of this document.

PARALLEL PROGRAMMING AND THE USE OF DIFFERENT EXTERNAL CRYSTAL FREQUENCIES (LPC21/22XX AND LPC17XX/40XX ONLY)

The parallel programming specification for LPC2000 family of microcontrollers contained in this document has been fully tested in systems running on an external crystal of 10 MHz only. However, the implemented on-chip ISP code makes provision for other crystal frequencies to be used, too. On the other hand, the parallel interface code (contained in the appendix of this specification) limits the values of external crystal to range from 10 MHz to 15 MHz.

The following table shows achievable UART0 baudrates versus the LPC2000 crystal frequency. For example, using an external crystal of 11.0592 MHz, the on-chip ISP can autobaud onto 9600, 19200, and 57600 bit/s baudrates.

Table 49: An overview of the ISP baudrates and selected external crystal frequencies (LPC21/22xx only)

ISP Baudrates .vs. External Crystal Frequency (in [MHz])	9600	19200	38400	57600	115200	230400
10.000	+	+	+			
11.0592	+	+		+		
12.2880	+	+	+			
14.7456	+	+	+	+	+	+

Messages and timings contained in this document have been specified for external crystal of 10 MHz. In case crystal value different from this one is used, message 3 in Table 24 and Table 48 will have to be changed as well as certain parameters required by chip erase and program procedures.

Changes in serial messages due to the external crystal frequency selection

The value supplied in message 3 (Table 24 and Table 48) equals the LPC2000 external crystal frequency measured in kHz. The following table lists content of message 3 for selected external crystal frequencies:

Table 50: Message 3 content for selected external crystal frequencies (LPC21/22xx only)

External Crystal Frequency (in [MHz])	Message 3 content matching the external crystal frequency
10.000	‘10000’+<CR><LF>
11.0592	‘11059’+<CR><LF>
12.2880	‘’12288’+<CR><LF>
14.7456	‘14745’+<CR><LF>

Other UART0 message contents are not affected by the change of LPC2000 external crystal frequency.

Changes in parallel communication procedures due to the external crystal frequency selection

Erase and Program Flash Memory procedures available through the parallel interface are dependent on the selection of the external crystal frequency.

Bytes 13 to 16 in Command 2, Erase Flash Memory Procedure and Bytes 17 to 20 in Command 2, Program Flash Memory and Verify Procedure should be changed in case external crystal frequency is altered. Value represented by these bytes is a hexadecimal equivalent of 5 times (LPC111x/11xxLV/11Cxx/11Dxx/11E11/11Uxx), 3 times (LPC12xx/12Dxx), 7 times (LPC1311/13/42/43/175x/176x), 10 times (LPC177x/8x and LPC407x/8x), 4 times (LPC21/22xx) or 7.2 times (LPC23/24xx) multiplied selected crystal frequency expressed in terms of kHz.

For example, in case of the external crystal frequency being 10 MHz, four times the crystal frequency produces 40 MHz. This value expressed in terms of kHz is 40,000 kHz and when converted into the hex format it becomes $40{,}000_{10}=9C40_{16}$. Since LPC1000/2000 microcontrollers are 32 bit devices, leading zeroes have to be added.

The following tables list external crystal frequency dependent fields in the above mentioned Command 2 in Erase and Program Flash Memory procedures:

Table 51: External Crystal Frequency dependant parameters for Flash erase and program procedures (LPC111x/11xxLV/11Cxx/11Dxx/11Exx/11Uxx/LPC1315/16/17/45/46/47 microcontrollers only)

External Crystal Frequency (in [MHz])	5 x External Crystal Frequency (in [kHz]) converted into hex
10.000	$5x10{,}000_{10}=50{,}000_{10}=C350_{16}$
11.0592	$5x11{,}059_{10}=55{,}296_{10}=D800_{16}$
12.2880	$5x12{,}288_{10}=61{,}440_{10}=F000_{16}$
14.7456	$5x14{,}745_{10}=73{,}728_{10}=12000_{16}$

Table 52: External Crystal Frequency dependant parameters for Flash erase and program procedures (LPC12xx/12Dxx microcontrollers only)

External Crystal Frequency (in [MHz])	3 x External Crystal Frequency (in [kHz]) converted into hex
10.000	$3x10{,}000_{10}=30{,}000_{10}=7530_{16}$
11.0592	$3x11{,}059_{10}=33{,}177_{10}=8199_{16}$
12.2880	$3x12{,}288_{10}=36{,}864_{10}=9000_{16}$
14.7456	$3x14{,}745_{10}=44{,}235_{10}=ACCB_{16}$

Table 53: External Crystal Frequency dependant parameters for Flash erase and program procedures (LPC1311/13/42/43 and LPC175x/6x microcontrollers only)

External Crystal Frequency (in [MHz])	7 x External Crystal Frequency (in [kHz]) converted into hex
10.000	$7x10{,}000_{10}=70{,}000_{10}=11170_{16}$
11.0592	$7x11{,}059_{10}=77{,}413_{10}=12E65_{16}$
12.2880	$7x12{,}288_{10}=86{,}016_{10}=15000_{16}$
14.7456	$7x14{,}745_{10}=103{,}219_{10}=19333_{16}$

Table 54: External Crystal Frequency dependant parameters for Flash erase and program procedures (LPC177x/8x and LPC407x/8x microcontrollers only)

External Crystal Frequency (in [MHz])	10 x External Crystal Frequency (in [kHz]) converted into hex
10.000	$10x10,000_{10}=100,000_{10}=186A0_{16}$
11.0592	$10x11,059_{10}=110,590_{10}=1AFFE_{16}$
12.000	$10x12,000_{10}=120,000_{10}=1D4C0_{16}$

Table 55: External Crystal Frequency dependant parameters for Flash erase and program procedures (LPC21/22xx microcontrollers only)

External Crystal Frequency (in [MHz])	4 x External Crystal Frequency (in [kHz]) converted into hex
10.000	$4x10,000_{10}=40,000_{10}=9C40_{16}$
11.0592	$4x11,059_{10}=44,236_{10}=ACCC_{16}$
12.2880	$4x12,288_{10}=49,152_{10}=C000_{16}$
14.7456	$4x14,745_{10}=58,980_{10}=E664_{16}$

Table 56: External Crystal Frequency dependant parameters for Flash erase and program procedures (LPC23/24xx microcontrollers only)

External Crystal Frequency (in [MHz])	7.2 x External Crystal Frequency (in [kHz]) converted into hex
10.000	$7.2x10,000_{10}=72,000_{10}=11940_{16}$
11.0592	$7.2x11,059_{10}=79,626_{10}=1370A_{16}$
12.2880	$7.2x12,288_{10}=88,473_{10}=15999_{16}$

For example, if a 14.7456 MHz crystal is selected for the programmed LPC21/22xx device and Erase Flash Memory Procedure is about to be performed, byte 13 of the Command 2 will be 0x64H, byte 14 will be 0xE6H, and bytes 15 and 16 will be 0x00H.

ENABLING CODE IN LPC1/2/4000 FLASH TO AUTO-RUN AFTER A RESET

In order to have code in LPC1/2/4000's Flash auto-run after a Reset, a 32-bit word of a specific value has to be programmed into the on-chip Flash at 0x0000_001C (LPC1000/4000 devices) and 0x0000_0014 (LPC2000 devices).

Here is the LPC1000/4000 procedure that determines what value to program:

1) make a 32-bit sum out of 32-bit words from Flash locations of 0x0000_0000, 0x0000_0004, 0x0000_0008, 0x0000_000C, 0x0000_0010, 0x0000_0014, and 0x0000_0018

2) find 2's complement of this sum

3) store this complement into 0x0000_001C

Here is the LPC2000 procedure that determines what value to program:

1) make a 32-bit sum out of 32-bit words from Flash locations of 0x0000_0000, 0x0000_0004, 0x0000_0008, 0x0000_000C, 0x0000_0010, 0x0000_0018, and 0x0000_001C

2) find 2's complement of this sum

3) store this complement into 0x0000_0014

Example 1: relationship between bytes and words in the on-chip Flash

If Flash location 0x0000_0000 contains 0x18, Flash location 0x0000_0001 contains 0xF0, Flash location 0x0000_0002 contains 0x9F and Flash location 0x0000_0003 contains 0xE5, than the 32-bit content residing at Flash address 0x0000_0000 has a value of 0xE59F_F018.

If the user specifies that the code that is about to be programmed into the LPC1/2/4000 device should auto-run after a Reset, than the right content for 0x0000_001C/0x0000_0014 can be calculated and inserted into the original binary content before actual code downloading into the microcontroller.

Example 2: calculating 32-bit word value at 0x0000_001C (LPC1000/4000 auto-run)

Here is a real-life application example:

Table 57: Flash memory locations (LPC1000/4000 devices)

Flash address	32-bit word content
0x0000_0000	0x1000_0B50
0x0000_0004	0x0000_0161
0x0000_0008	0x0000_0165
0x0000_000C	0x0000_0167
0x0000_0010	0x0000_0169
0x0000_0014	0x0000_016B
0x0000_0018	0x0000_016D
0x0000_001C	undefined

According to description from the above, a 32-bit sum is calculated. The 32-bit value corresponding to data provided in Table 58 equals 0x1000_13BE.

One number's 2's complement is calculated as this number's 1's complement incremented by 1. In the case illustrated with Table 57, 1's complement of the 32-bit sum is 0xEFFF_EC41. Incrementing this by 1 gives 0xEFFF_EC42.

Therefore, programming a 32-bit word of 0xEFFF_EC42 in this example as of Flash address 0x0000_001C will enable the LPC1000 device to run the application residing in its Flash memory after a reset.

Example 3: calculating 32-bit word value at 0x0000_0014 (LPC2000 auto-run)

Here is a real-life application example:

Table 58: Flash memory locations (LPC2000 devices)

Flash address	32-bit word content
0x0000_0000	0xE59F_F018
0x0000_0004	0xE59F_F018
0x0000_0008	0xE59F_F018
0x0000_000C	0xE59F_F018
0x0000_0010	0xE59F_F018
0x0000_0014	undefined
0x0000_0018	0xE51F_FFF0
0x0000_001C	0xE59F_F018

According to the step 1) from the above, a 32-bit sum is calculated. The 32-bit value corresponding to data provided in Table 58 equals 0x46DF_A080.

One number's 2's complement is calculated as this number's 1's complement incremented by 1. In the case illustrated with Table 58, 1's complement of the 32-bit sum is 0xB920_5F7F. Incrementing this by 1 gives 0xB920_5F80.

Therefore, programming a 32-bit word of 0xB920_5F80 in this example as of Flash address 0x0000_0014 will enable the LPC2000 device to run the application residing in its Flash memory after a reset.

APPENDIX

Application Code for Parallel Bootloader/Parallel Interface (Revision 2.0) - UART0 download

Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

This code can be used by LPC21xx/22xx devices only!

Binary set of data that follows message 5 is 720 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends to 16*61 characters. The following is an ASCII dump of the content a programmed LPC2000 device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 1073742336 720
MF"*?Y9@2G^4(('E_RR@XPP@@>4``-#E``2@X00`@>4``.#A#`"!Y00PH.,,
M,('E``"1Y8``$./\__\*#""!Y7X`H.,(`('E!#"!Y1[_+^'X3RWI3+*?Y412
MG^4"8*#C!&"%Y0"`H./H(N#C"S"@X0MPH.%(3HOB4,Z+X@2@H.,``)7E`0`0
MXR(``!H``)7E`!"5Y8```.(!$!'B`>"@`P#@H!,``%#C`0``"@``4>/U__\*
M``!>XQ0```H`$)7E`1B@X2&<H.$!@(CB#*"%Y60`6>,!`%@#!0``&@`0H.,$
MX)/D`1"!X@0`4>,$X(SD^O__V@&0Q^0``%#C``"5Y8````+[__\*`1``X@2@
MA>4``%'CV?__"@!PF^4+`*#A@`]7XT`/5Q/`#U<3#```"@00H.$/X*#A$O\O
MX3<`5^.`'Z`#"!"$!0``E.6``:#A>!``X@00A>5X$*#C``#!X0P`A>4,8(7E
M^$^]Z![_+^$T`9_EJA"@XXP0@.55$*#CC!"`Y1[_+^'X3RWI'$&?Y10AG^4C
M`*#CA`""Y0%PH.-(CD3B@%"$XO#__^L!,*#C@#""Y>W__^N(`)+E0`X0X_S_
M_PH#`*#C@`""Y>?__^M`,(+EV!"?Y0``H.,``('E?@"@XP"4P>$(`(GE!`"@
MXP0`B>5X`*#C#`")Y;2@G^6`L$KB``"9Y0`0F>4!``#B@!`!X@``4.,"```*
M``!1X_?__QH!``#J``!1XP$``!H``)GE`0``X@``4.,$```:"T"@X0I0H.$!
M<*#C?___Z^O__^H``)CE0`]0XP%0A0(%`*`!"P``"H`/4.,!0(0"!`"@`0<`
M``K`#U#CX/__&@``5^,$$)@5`'"@XP%@01(!8(;B!@"@X5C__^O8___J?O\`
M``"``N``!0!``,`?X(`)`$``P`+@_PD`0`30G^4$X)_E#O"@X?P/`$"(`P!`
78719
```

Application Code for Parallel Bootloader/Parallel Interface (Revision 3.0) - UART0 download

This code can be used by LPC23xx/24xx devices only! Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

Binary set of data that follows message 5 is 1076 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends to almost 24*61 characters. The following is an ASCII dump of the content a programmed LPC2000 device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 1073742336 1076
M!/`?Y0@"`$`@`)_E$/`AXP#0H.$8`)_E`0`0XQ3@GP44X)\5$/\OX?[__^K^
MY\!&``<`0-T%`$`H`@!`+0(`0`````````````````````#PM86P`"(`D@$@
M2DD$D4I)`I%)24E,2DH#D@(C24H38`$E*QQ(2A)H`28R0!8<`"X!T``@`.`H
M'`W@0TH3:$-*$T!!2A)H`24J0!4<`"T!T``@`.`!(!T<`"T"T!4<`"WKT`4<
M`"TUT#E-+FC_)2Y`-@8V#@&M+G``G@$V`)8U3S5.-V`M>&0M#]$`G0$M#-$`
M)0X<-V@#GC=@`YX$-@.6!#$!-2X<!"[SW0&M+G@"G2YP`34"E0/@(TL;:"--
M*T`='``M^-`@2A)H`2,:0"%-'$L=8!,<`"NBT!=)"!P%:"@<'$J00B#0*!P8
M2I!"'-`H'!I*D$(8T`@<(1P$F@#P:_DH'#<H!-$(-!1)(!P!8`@\(!P":!(!
M$QSP(0M`"4@#8!`<@4,,2`%@`B$*2`%@!;#PO?'__W\`!P!`0`L`0(`+`$`8
MP/\_%,#_/P`!``!4P/\_``(``!S`_S\``P```AP`(?\A#D@!8`Y(`6`0'`!X
M`1P,2`%@#$D-2`%@#4@!:`U(`4`('``H^-#_(05(`6``(0)(`6`$20A(`6!P
M1T#`_S]<P/\_5,#_/P`"```<P/\_%,#_/P`!```8P/\_\+6!L`$A5$@!8*HA
M4T@!8%4A4D@!8%)(`6@P((%#("`!0T](`6!.2`!H0"$(0OK0`2%,2`%@2$@!
M8*HB1T@"8%4A1D@!8$A+2$@#8$-(`F!"2`%@`R%&2`%@1D@`:$9)"$+ZT`,A
M/$@!8*HA.T@!8%4A.D@!8#I(`F@!(0I#.$@"8#Y*/D@"8#Y*/T@"8#](`F@(
M(()#/4@"8#U*/4@"8``B/4@"8`(B/$@"8#Q*/4@"8/`B/$@"8#Q//4@`D#U.
M`24]2`%@`2,9'#Q(`6@!(`%`.DH3:#I*$T`*'``J`M`:'``J\=$:'``J`M$S
M20EH`4`('``H!]$L3P$_+4X!/@$E__>=_M[G`)L8:"U)"1@*T"U)"1@,T!U)
M"1C3T0$V,!S_]SW_SN<!-S@<__<X_\GG*!P`*`?0&!P$,`"0!&@!/`"8!#@`
MD`$T`"4@'/_W)_^XYP&P\+V`P!_@C,`?X*#!'^`,P1_@1P`$`(3`'^`$P1_@
64425
MB,`?X`````0,_/__$,#_/_("````P/\_*,`"X`#___]0P/\_0,#_/QS`_S\`
M`@``&,#_/US`_S]`"P!```<`0(`+`$!`P!_@%,#_/P`!````_O__`/W__P](
M!L@+'!-#!=``(PMP21R10OO1]><+2,`<`R.80P;("QP30P;0`WA`'`MP21R1
I0OG1\><`M?_W_?[^YP"]P$9``@!`2`(`0`!'"$<01QA'($<H1S!'.$='
17594
```

Application Code for Parallel Bootloader/Parallel Interface (Revision 4.0) - UART0 download

This code can be used by LPC175x/6x devices only! Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

Binary set of data that follows message 5 is 1152 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends to almost 26*61 characters. The following is an ASCII dump of the content a programmed LPC17xx device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 268436256 1152
M0!``$&4$`!!I!``0:P0`$&T$`!!O!``0<00`$`````````````````````!S
M!``0=00`$`````!W!``0>00`$'L$`!![!``0>P0`$'L$`!![!``0>P0`$'L$
M`!![!``0>P0`$'L$`!![!``0>P0`$'L$`!![!``0>P0`$'L$`!![!``0>P0`
M$'L$`!![!``0>P0`$'L$`!![!``0>P0`$'L$`!![!``0>P0`$'L$`!![!``0
M>P0`$'L$`!![!``0>P0`$`#P`O@`\"[X"J"0Z``,@D2#1*KQ`0?:10'1`/`C
M^*_R"0ZZZ`\`$_`!#QB_^QI#\`$#&$=X`P``B`,````C`"0`)0`F$#HHOWC!
M^]A2!RB_,,%(OPM@<$<?M:_S`(`?O1"U$+T`\&?Y$4;_]_7_`/!;^0#PA?G_
M]_/_`/"+^0``"T@`1_[G_N?^Y_[G_N?^Y_[G_N?^Y_[G!D@'205*!TMP1P``
M!DC0^`#0!DC0^`#PY0,`$*`(`!!`$``00`X`$"`#`!`D`P`0$+612?\BBF4`
M(\ME#&Q$\/\$#&0$>(QE`'C`0\"RR&6`),QA2&G`\X`0`"CZT(IERV4(;"#P
M_P`(9(QA$+TMZ?!'@4V!3H))@DQ/\`()Q?@8D(`C:&D0\`$/'=%H:6IIP/.`
M$!+P`0(!T``G`.`!)PBQ`"KRT&^Q:FWK80'X`2L"X&AIP/.`$``H^M!H::MA
M`/`!`@`JW=!M2`=HM_4`?R'03_``"+?U0'\7T"%&L$<U+PO0(&@(*``1Q/@`
M@#<O!-%/](!@1/@(#P@\(&@!!PD-J6$/(8%#"`(#X,7X&(!/]'!@Z&'%^!R0
MO>CPARWI\$%92@`F$&BJ)"#P`@`08-1@52#08#%&%F#48-!@("/"^"`QLO&`
M`M+XH#&;"=L'^M`!(X`RPOB,,`TE56#48-!@$V#48-!@`R7"^(10@#K2^(AP
MOP[_!_K00OB`7]1@T&"`.M+XB`!`#L`'^M`_3"%@86"A8"%D862A9#9(`6`!
M8@%D)6Q%\`(%)60E;$_P`@Q%\`@%)63`^!C`!6A%\`(%!6`E;$7T*@4E9(%A
M3_1P:,#X'(`%:$7T<&4%8"5L1?0`525D)6Q/\(`.1?0`125DP/@8X`5H1?"`
M!05@)6U*]JHG/4,E91Y-'4\!)!-DP/@<P,#X&."!8<#X'("`1D)&4&E1:1#P
81357
M`0#!\X`1`M``*?;1`.`1N5!I`/`!`"BY$$T!)&T>__<2_^GG.&BP]0!_#M"P
M]4!_XM$<L5?X!&]V'C\?=AP`)$_J!@#_]^#^UN=M'"A&^><`P`D@\1__'P`*
M`!``"``0@,`/0`#``D`0M?_W1/_^YW!'<$=P1W5&`/`H^*Y&!0!I1E-&(/`'
M`(5&&+`@M?_WH_Z]Z"!`3_``!D_P``=/\``(3_``"R'P!P&L1JSHP`FLZ,`)
MK.C`":SHP`F-1G!'!$:O\P"`($;_]W7^`$AP1T`(`!`#24_P&`"KOG!'```)
;````)@`"`'!'``"@!P`0``@`$$`(```@!``0
24686
```

Application Code for Parallel Bootloader/Parallel Interface (Revision 5.0) - UART0 download

This code can be used by LPC1311/1313/1342/1343 devices only! Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

Binary set of data that follows message 5 is 1220 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends to almost 28*61 characters. The following is an ASCII dump of the content a programmed LPC1311/1313/1342/1343 device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 268436256 1220
MH`D`$!$'`!`5!P`0%P<`$!D'`!`;!P`0'0<`$`````````````````````````?
M!P`0(0<`$``````C!P`0)0<`$`````````````````````````````````````
M`````````````````````````````````````````````````````````````
M`````````````````````````````"<'`!`G!P`0)P<`$"<'`!`G!P`0
M)P<`$"<'`!`G!P`0)P<`$"<'`!`G!P`0)P<`$``````G!P`0)P<`$"<'`!`G
M!P`0`/`"^`#P+O@*H)#H``R"1(-$JO$!!]I%`=$`\"/XK_()#KKH#P`3\`$/
M&+_[&D/P`0,81YP#``"L`P```",`)``E`"80.BB_>,'[V%('*+\PP4B_"V!P
M1Q^UK_,`@!^]$+40O0#P>?D11O_W]?\`\$GY`/"7^?_W\_\`\)WY```PM91+
M0/(_,L/X_"R23"%H$4,A8`!X`/#``0#P/P``ZX$`P_C\#(U-`"`H8(Q)`+_1
M^/P/P/.`(``H^=##^/PL(&B00R!@3_0`8"A@,+TMZ?!'W_@0@H1)A$R`(D_P
MH$G)^``B?D^"37Q+`";5^/P/P/.`$(B[`+_7^/P/U?C\+\#S@"#"\X`2$K%/
M\``,`>!/\`$,"+$`*N[0O/$`#QK0U_C\+]?X_,\"\#\"#/1`?`+KG`(>8`'X
M`2L#X-?X_`_`\X`@`"CYT-7X_"]/]`!@PO.`$AA@`"K(T&-(!VBW]0!_&="W
M]4!_%-`A1L!'-2\*T"!H""@`T29@-R\$T4_TH&!$^`@/"#P@:``'``W%^``,
M`>#%^`!LR?@`8KWH\(<MZ?!'4TD`)@AN4TH00PAF,D8.8-'X&#(P1B/P(`/!
M^!@R(#E`'/\H_-D!(PMD2V1*9$MD2&S`!_S0)B"(8-'X.`(@\(``P?@X`LAH
MP`?\T`,@"&=+9TIG2V=(;\`'_-"+9SY(0".#8L-BPV,#9$$D1&3!)`1E1&6$
M9<1EQ&;`)`1GPV=`^!`\0V&$9$1G-$L:8"E(`F`$:$#R/S6L0P1@'&A$]'AD
M'&`#:"A-0_0`8P-@`20D3P(@"&!/\*!`P/@`(AY)X`((8"%(P/@`+-_X<)"`
M1DM&0D;2^/P/T_C\'\#S@!#!\X`A$+$`*?31`.`9N=+X_`_`\X`0*+D330$D
75660
M;1[_]Q3_Y><X:+#U`'\.T+#U0'_>T1RQ5_@$;W8>/Q]V'``D3^H&`/_WVO[2
MYVT<*$;YYP```5``@`%0`"`!4``P`5#Q'_\?``H`$``(`!``,`!0((`$0$``
M`0`D0`1``(``4!"U__=+__[G"T@`1_[G_N?^Y_[G_N?^Y_[G_N?^Y_[G!D@'
M205*!TMP1P``!DC0^`#0!DC0^`#P!00`$*`(`!"@"0`0H`@`$"`#`!`D`P`0
M<$=P1W!'=48`\"CXKD8%`&E&4T8@\`<`A488L""U__?7_[WH($!/\``&3_``
M!T_P``A/\``+(?``':Q&K.C`":SHP`FLZ,`)K.C`"8U&<$<$1J_S`(`@1O_W
M8_X`2'!'0`@`$`-)3_`8`*N^<$<```D````F``(`<$<``.0'`!``"``00`8`
%`$`$`!`0
30460
```

Application Code for Parallel Bootloader/Parallel Interface (Revision 6.0) - UART0 download

This code can be used by LPC111x/11Cxx11/Dxx devices only! Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

Binary set of data that follows message 5 is 664 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends to almost 15*61 characters. The following is an ASCII dump of the content a programmed LPC111x/11Cxx/11Dxx device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 268436544 664
M`4B%1@%(`$?@!P`0400`$`-(A48`\!;Y`$@`1UL%`!#@!P`0<+5\2GI)$6-[
M2QE@`'B$!X`(@``D#P`!`!D08W=-`"0L8'1.P#;P:X`&P`_[T!%C'&`0("A@
M<+WXM7%(("*$1FY(<$EP3$`P`F!J2VM/P#,`)=AKP`;`#R?1V&O::X`&P`_2
M!M(/`=``)@#@`28`*``0`"KQT``N%=#>:_PB-@D60-IK/6`2!Y(/LA@*<$D<
M`N#8:X`&P`\`*/K0VFL0(-(&T@\X8``JT]!62%-/!F@#(O$?0#\2`O\Y^CD4
MT)9"$M`A1F)&D$<U+@C0(&@(*`#1)6`W+@+1`R!``J!@(&@`!P`/^&,"X)9"
M`-']8T-(0#`%8/B]^+5$20`F"&A$2A!#"&!!2#-&@#@&8D%/N6L@(I%#N6,Q
M1DD<_RG\V3M)`25`.0U@36!+8$U@2FC2!_S0)"*"8+IK@"2B0[ICPFC2!_S0
M`R(*8TUC2V--8TIKT@?\T(UC,$K1(9%B46.18]%C+4I`,A%@$6$C20M@*TH_
M(1%@`20D32)/`B$!8!](0#`#8!U)$"`(8!Q(0#C#8QA*P#+0:]%KP`:)!L`/
MR0\`*`+0`"GUT0'@`"D"T=!KP`;`#P`H!=$330$D;1[_]SO_Y><X:/\X_S@"
M.`S0_S@!.-W1`"P!T'YH=AYV'``D,$;_]PW_T^=M'"A&^><``,P/````/P!0
M`(``4$```5#Q'_\?````$-@&`!"`@`1`0``!``""!$!`0`1``(`!4`9,`24&
M3@7@($;C:`?(*T.81Q`TM$+WT__WW/[(!@`0V`8`$`+@",@2'PC!`"KZT7!'
B<$<`(`'@`<$2'P`J^]%P1P``V`8`$-@&`!`(`0``N`8`$!`0
56890
```

Application Code for Parallel Bootloader/Parallel Interface (Revision 6.1) - UART0 download

This code can be used by LPC11Uxx/11Exx/1315/1316/1317/1345/1346/1347 devices only! Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

Binary set of data that follows message 5 is 608 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends to almost 14*61 characters. The following is an ASCII dump of the content a programmed LPC11Uxx device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 268436544 608
M`4B%1@%(`$?@!P`0400`$`-(A48`\!+Y`$@`1UT%`!#@!P`0$+5Y20IH_R.;
M`1I#"F!W2A1H'$,48`QH`'B<0X`!!$,,8'-,!"`@8`QHI`;D#_O0#&@<0PQ@
M$6B90Q%@;$F`.0A@$+WXM0$@:4D`!6E*:DMJ38`Y"&!D3`0G(&C`!L`/(M$@
M:"%H@`;`#\D&R0\!T``F`.`!)@`H`=``*?'0`"X0T"%H6DZ)"3=@&7!;'`'@
M@`;`#P`H(&CZT,`&P0]42(`X!V``*=C04T@#(09H"0+S'_\[^CL:T`\GOP..
M0A/0*4:01S4N"-`H:`@H`=$`("A@-RX!T4I(J&`H:"%H@`.Y0P%#(6`"X"!H
MN$,@8`$A/T@)!0%@^+U"2``D`6A"2A%#`6`_22-&@#D,8C]-J&L@(I!#J&,@
M1D`<_RC\V3E(`2)`.`)@0V!"8*YK@">^0ZYC)"6-8,UH[0?\T`,E!6-#8T)C
M@F,R2I$@D&+08A!C4&.08]!C)4HN2!!@`2<G325.`B`(8"))P`0(8"!)!""`
M.0A@'$@!:`\BD@.10P%@&4D(:,`&P@\(:(`&P`\`*@+0`"CTT0'@`"@8T1)(
M`&C`!L`/$]`P:/\X_S@".!30_S@!..31`"\!T'1H9!YD'``G($;_]QS_`2(0
M1M_G"TT!)VT>__<V_P/@;1PH1O_W#_\!(-7G````(0!0`"``4(`B`%#Q'_\?
M````$``'`!`!!@``@(`$0$```0``@@1``$`$0`3`$P`&3`$E!DX%X"!&XV@'
7R"M#F$<0-+1"]]/_]^#^H`8`$*`&`!`0
48402
```

Application Code for Parallel Bootloader/Parallel Interface (Revision 6.2) - UART0 download

This code can be used by LPC11Axx devices only! Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

Binary set of data that follows message 5 is 608 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends a little more than 13*61 characters. The following is an ASCII dump of the content a programmed LPC11Axx device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 268436544 588
M`4B%1@%(`$?@!P`0400`$`-(A48`\`CY`$@`1U\%`!#@!P`0$+5U20IH_R,;
M`1I#"F!S2A1H'$,48`QH`'B<0P`!!$,,8`$@;DS``R!@#&@D!^0/^]`,:!Q#
M#&`1:)E#$6!H28`Y"&`0O?BU`2!E28`#94IE2V9-@#D(8%],1P`@:$`'P`\B
MT2!H(6@`!\`/20?)#P'0`"8`X`$F`"@!T``I\=``+A#0(6A63@D)-V`9<%L<
M`>``!\`/`"@@:/K00`?!#T](@#@'8``IV-!/2`,A!F@)`O,?_SOZ.QK0#R>_
M!(Y"$]`I1I!'-2X(T"AH""@!T0`@*&`W+@'114BH8"AH(6B`!+E#`4,A8`+@
M(&BX0R!@`2$[2(D#`6#XO3Y(`"0!:#U*$4,!8#U(@FN`(8I#@F,X2B,@@#J0
M8-!HP`?\T#5(`R-`.`-C`"-#8P$C0V.#8S-(@6#!8)$A06&!8<%A`6)!8H%B
M\R`E28`#"&`!)R=-)4X"(!!@(DE``PA@($E``(`Y"&`<2`%H#R*2!)%#`6`9
M2A!H0`?!#Q!H``?`#P`I`M``*/31`>``*!C1$D@`:$`'P`\3T#!H_SC_.`(X
M%-#_.`$XY-$`+P'0=&AD'F0<`"<@1O_W)/\!(0A&W^<+30$G;1[_]S__`^!M
M'"A&__<7_P$@U><````A`%``(`!0@"(`4/$?_Q\````0``<`$`(&``"`@`1`
M0``!``""!$``0`1`!DP!)09.!>`@1N-H!\@K0YA'$#2T0O?3__?J_HP&`!",
#!@`0
44946
```

Application Code for Parallel Bootloader/Parallel Interface (Revision 6.3) - UART0 download

This code can be used by LPC1111/1112 20-pin devices and LPC1101LV/1102LV/1112LV/1114LV devices only! Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

Binary set of data that follows message 5 is 684 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends a little more than 15*61 characters. The following is an ASCII dump of the content a programmed LPC1111/1112 20-pin device or a LPC1101LV/1102LV/1112LV/1114LV device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 268436544 684
M`4B%1@%(`$?@!P`0400`$`-(A48`\#CY`$@`1VD%`!#@!P`0\+4/(89,"0(A
M8(9.#R7U8X5*$V@+0Q-@A$L?:"]#'V`'>*]#/P$G8`!X*$#P8X!(`"<'8']/
M_VM_!O\/^M`A8/5C%&B,0Q1@&6@)"0D!&6"`(0%@\+WXM7=(!"&$1@4@=DMW
M3``'`6%V2@`ET&N`!\`/+-%O3O!K0`;!#]!K@`?`#P'0`"<`X`$G`"D!T``H
M\=``+QG0T&OP)P`).$`W1O9K-@<V#S!#84XU8!AP6QPX1@+@P6M)!LD/`"GZ
MT-!K6TF`!\`/@"8.8``HSM!;2`9H\1__.?HY$-`#(0D"CD(,T"%&8D:01S4N
M!]`@:`@H`-$E8#<N`=%32*!@!2``!P5A^+U12``D06E)!HD/`RGZT4](@&@`
M"0#@0!X`*/S13$@!:$Q*$4,!8$I(`"&`.`%B2DZR:R`CFD.R8PI&4AS_*OS=
M0TH!)4`Z%6!18%5@MVN`(Y]#MV,D)H9@QFCV!_S0`R868U%C56.58SQ*P";6
M83M/@#=^8KYB.4W!(D`UJF/J8SI@.F$N8FYBJF)J8R5*%6@$)C5#%6`D2A5H
M'4,58`$G)DTE3@(B`F`%(``'`6$?2`-@(TK0:X`'P0\=2,!K0`;`#P`I`M``
M*//1`>``*!C1&TC`:X`'P`\3T#!H_SC_.`(X%-#_.`$XX]$`+P'0=&AD'F0<
M`"<@1O_W`/\!(0A&W^<.30$G;1[_]R/_`^!M'"A&__?S_@$@U><````\`%``
M``%0`(``4`"``5```@%0P#\!4/$?_Q\````0``<`$,`_`%`#!@```(``0"`
M`4"`@`1`0``!``""!$``0`1`!DP!)09.!>`@1N-H!\@K0YA'$#2T0O?3__>Z
)_NP&`!#L!@`0
56512
```

Application Code for Parallel Bootloader/Parallel Interface (Revision 7.0) - UART0 download

This code can be used by LPC1112 24-pin devices only! Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

Binary set of data that follows message 5 is 644 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends to almost 15*61 characters. The following is an ASCII dump of the content a programmed LPC1112 24-pin device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 268436544 644
M`4B%1@%(`$?0!P`0400`$`-(A48`\`SY`$@`1U\%`!#0!P`0\+4/(7-*"0(1
M8'-+#R7=8W),(6#_)G),(#8F8`9XKD,V`19@!G@`("Y`WF-M3C!@;4__:S\&
M_P_ZT!%@W6-G20A@_R`1,"!@_R`!,#!@\+WXM6%($"%E2F5-9DQ`,`%@`"9A
M2,%K20;)#RO1PVO!:QL&VP])!LD/`=``)P#@`2<``*P'0`"GQT``O&=!62,!K
M64GP)P`)R6LX0`D'44\)#PA#/F`H<&T<3T@!X`D&RP_!:P`K^M#_(`$P20;)
M#SA@`"G.T$I(!6CI'_\Y^CD0T`,A"0*-0@S0(4:01S4M"-`@:`@H`-$F8#<M
M`M$'(``"H&`Y2$`P!F#XO4!(`"4!:#]*$4,!8#U)(R"`.8A@/4B":X`CFD."
M8\AHP`?\T#=(`R)`.`)C`2-#8P`B0F-#8T1KY`?\T(-C,TO1()AB6&.88]AC
M,$M`,QA@&&$C2`)@_R`C2Q$P&&`!)R9,)$X"(`A@'4A`,`)@'DDX`@A@'DG*
M:\AK4@;2#P`&P`\`*@+0`"CTT0'@`"@8T1=(P&M`!L`/$]`P:/\X_S@".!30
M_S@!..31`"\!T'5H;1YM'``G*$;_]R7_`2(01M_G#4P!)V0>__=%_P/@9!P@
M1O_W&/\!(-7G`#P`4````5``@`!0`(`!4``$`5#`/P!0\1__'P```!#$!@`0
MP#\!4("`!$!```$``(($0$!`!$`&3`$E!DX%X"!&XV@'R"M#F$<0-+1"]]/_
M]^;^M`8`$,0&`!`"X`C($A\(P0`J^M%P1W!'`"`!X`'!$A\`*OO1<$<``,0&
.`!#$!@`0#`$``*0&`!`0
52599
```

Application Code for Parallel Bootloader/Parallel Interface (Revision 8.0) - UART0 download

This code can be used by LPC12xx/12Dxx devices only! Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

Binary set of data that follows message 5 is 636 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends to more than 14*61 characters. The following is an ASCII dump of the content a programmed LPC12xx/12Dxx device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 268436544 636
M`4B%1@%(`$?(!P`0400`$`-(A48`\`CY`$@`1TD%`!#(!P`0$+4%(0D'3&@_
M(A(!_R,40)L#Y!B,8'%,#&),:`!X%$"``P1#C&`@(`AA3&@D!^0/^]!,:!1`
MXQB+8`IBR&`0O?BU$"`%)&9*9DMG320'X&`@)V!HP`?`#Q_18&AA:``'P`_)
M!\D/`=``)@#@`28`*``0`"GQT``N#=!A:"=AB0L9<%L<`>``!\`/`"A@:/K0
MP0?)#^=@`"G;T%)(`R$&:`D"\Q__._H[&=!03XY"$]`I1I!'-2X)T"AH""@!
MT0`@*&`W+@+1`2#``JA@*6A@:(D!.$`(0P'@8&@X0*!@$"`@8?B]0D@`)4%I
M20:)#P,I^M%`2!\A`6`!:`$BT@,10P%@/4D`(PM@JB2,8%4DC&`!:)%#`6`!
M:#A*$4,!8#5)0B"`.8A@-DB":X`DHD."8\AHP`?\T"](`R)`.`)C`2)"8T-C
M0F-#:]L'_-""8S\B$@$%(``'`F(!)R-,(4X"(@I@$"$!82`AP6!!:!]*$4"!
M8`4B$@=1:%!HR0?)#P`'P`\`*0+0`"CST0'@`"@9T04@``=`:,`'P`\3T#!H
M_SC_.`(X%-#_.`$XXM$`+P'0=6AM'FT<`"<H1O_W(?\!(0A&WN<(3`$G9![_
M]SG_`^!D'"!&__<4_P$@U.?PPS\`\1__'P```!"\!@`0,,`_``"``$"`@`1`
M`$``0````8``@@1`!DP!)09.!>`@1N-H!\@K0YA'$#2T0O?3__?J_JP&`!"\
M!@`0`N`(R!(?",$`*OK1<$=P1P`@`>`!P1(?`"K[T7!```"\!@`0O`8`$`P!
&``"<!@`0
50915
```

Application Code for Parallel Bootloader/Parallel Interface (Revision 9.0) - UART0 download

This code can be used by LPC177x/8x and LPC407x/8x devices only! Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

Binary set of data that follows message 5 is 680 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends to more than 15*61 characters. The following is an ASCII dump of the content a programmed LPC177x/8x or LPC407x/8x device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 268436480 680
M!$@%20-*!4MP1P``!$B%1@1(`$>@"``0H`D`$*`(`!"@"0`0*00`$*_S`(``
M\`;X'[4`\#+Y'[T0M1"]`/#S^!%&__?U_P#P9O@`\!'Y`[3_]_+_`[P`\!GY
M``!J2?\B2F4*9`!X2&6`(,AA2VG#\X`3`"OZT$IE`"(*9(AA<$<MZ?!!8$QA
M2F%+8DT")Z=A3_"`#&!I$/`!#Q[18&G`\X`18&D0\`$``=``)@#@`28)L0`H
M\M!VL6!MQ/@<P`/X`0L!X,#S@!%@:0`I^M``\`$`Q/@8P``HW-!-2`9HMO4`
M?Q70MO5`?P[0*4:01S4N"=`H:`@H`=$`("A@-RX"T4_T$&"H8"AH8&D@]'!@
M8&'G8;WH\(%`2$_T`$$`)<#XQ!`@(L#XH"'0^*`A4@;[U0#Q@``#(L#X,"$X
M2L#X""%!9$_P`0+`^(P@`F!/\`D!06!/\*H!P6!/\%4!P6"@\8```+_0^(@0
M207[U<#XJ"'`^`0A0/(!$<#X!!$B20`C"V1`]H)S"V`B3"%.`2<"9`(CRV&`
M(XMA2VDC]'!C2V'`^*@A3^H!"-CX%```\`$"V/@4`,#S@!`2L0`H]-$`X+"Y
MV/@4`!#P`0\1T#!HL/4`?Q/0L/5`?^;1#[%U:&T>;1P`)RA&__<Y_P$B$$;C
MYPA,`2=D'O_W0_\#X&0<($;_]RS_`2#8YP"`"2#Q'_\?``H`$``(`!``P`]`
M!``!`'!'<$=P1W5&`/`H^*Y&!0!I1E-&(/`'`(5&&+`@M?_WW_Z]Z"!`3_``
M!D_P``=/\``(3_``"R'P!P&L1JSHP`FLZ,`)K.C`":SHP`F-1G!'!$:O\P"`
M($;_]^G^`$AP1T`(`!``2'!'1`@`$`%)&""KOO[G)@`"`'!'$+7_]_+_3_``
%`0%@$+V]
69979
```

Application Code for Parallel Bootloader/Parallel Interface (Revision 10.0) - UART0 download

This code can be used by LPC181x/2x/3x/5x and LPC431x/2x/3x/5x devices only! Parallel Bootloader Code revision can be obtained as CodeVersionP using “Read Boot Code“ procedure.

Binary set of data that follows message 5 is 568 bytes long. However, in order to be transmitted over the UART0, it is converted into the UUENCODED format, and then it expends to almost 14*61 characters. The following is an ASCII dump of the content a programmed LPC181xx/43xx device would receive over the UART0 channel. It includes message no. 5 and subsequent UUENCODED data:

```
W 268436544 568
M`4B%1@%(`$?@!P`0400`$-_X#-``\/[X`$@`1T\%`!#@!P`0=$G_(L'XA"%+
M:$/P_P-+8`!XP?B$`0@@P?B,`M'XE#'#\P`3`"OYT,'XA"%*:"+P_P)*8,'X
M#`)P1RWI\$%F2`)H9$QE2V9-3_`!",3X#()/\`@'U/B4`<#S@``@NP"_U/B4
M`=3XE!'`\P`0P?.``0FQ`"8`X`$F"+$`*?#0CK'4^(01Q/B,<@/X`1L"X`"_
MP/,`$``HU/B4`?G0P/.``<3X#'(`*=703$@&:+;U`'\7T``GMO5`?Q'0*4:0
M1S4N"-`H:`@H`-$O8#<N`M%/]"!@J&`H:$`!Q/B4`0'@Q/B4<<3XC(*]Z/"!
M/D@`)$%IP?-!$0,I^M%`]L000![]T3E)<"#!^&@"P?AT`D_P=`+*9D_T^G-+
M9PA@2&"(8,A@"&%(88AAR&&*9\IGP?B$(,'XZ"`M2"I-*$Y/\``!3_`!!\#X
M`!$C2,#XC')/\`@"P/@,(L#XE!'!:$'P`0'!8,%H0?`(`<%@06E!]/!Q06%/
MZ@`(V/B4`<#S@`+8^)0!P/,`$!*Q`"CTT0#@L+G8^)0!P/.``(BQ,&BP]0!_
M$]"P]4!_YM$/L71H9!YD'``G($;_]R7_`2(01N/G"$T!)VT>__<Z_P/@;1PH
M1O_W&/\!(-CG`&`/0``!0!`````0``<`$``0"$"<8`A``#`$0`9,!TT&X.!H
<0/`!`Y3H!P"81Q`TK$+VT__W]/YX!@`0>`8`$!`0
62389
```

STATUS LINE CODES

Table 59: Status codes

Status Code	Description	Mnemonic
00H	Command is executed successfully.	CMD_SUCCESS
01H	Invalid command.	INVALID_COMMAND
02H	Source address is not on a word boundary.	SRC_ADDR_ERROR
03H	Destination address is not on a correct boundary.	DST_ADDR_ERROR
04H	Source address is not mapped in the memory map. Count value is taken in to consideration where applicable.	SRC_ADDR_NOT_MAPPED
05H	Destination address is not mapped in the memory map. Count value is taken in to consideration where applicable.	DST_ADDR_NOT_MAPPED
06H	Byte count is not multiple of 4 or is not a permitted value.	COUNT_ERROR
07H	Sector number is invalid.	INVALID_SECTOR
08H	Sector is not blank.	SECTOR_NOT_BLANK
09H	Command to prepare sector for write operation was not executed.	SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION
0AH	Source and destination data is not same.	COMPARE_ERROR
0BH	Flash programming hardware interface is busy.	BUSY
0CH	Insufficient number of parameters or invalid parameter	PARAM_ERROR
0DH	Address is not on word boundary	ADDR_ERROR
0EH	Address is not mapped in the memory map. Count value is taken in to consideration where applicable.	ADDR_NOT_MAPPED
0FH	Command is locked.	CMD_LOCKED

UART0 SERIAL COMMUNICATION RESPONSE CODE SUMMARY

During regular communication between a parallel programmer and a LPC1000/2000 device over the UART0 response code “0“ (command successfully executed) will be sent. An even when any other code is noticed indicates that something is in the process is not aligned with the parallel programming specification. Problems may be related to the part that is programmed, as well as to the format and content of messages the programmer has been sending to the LPC1000/2000 device. For more details on reported errors, please use the corresponding LPC1000/2000 microcontroller user manual.

Table 60: UART0 Serial Communication Response Codes

Response Code (hex/dec)	Description	Mnemonic
00H/ 0	Command is executed successfully. Sent by ISP handler only when command given by the host has been completely a nd successfully executed.	CMD_SUCCESS
01H/ 1	Invalid command.	INVALID_COMMAND
02H/ 2	Source address is not on word boundary.	SRC_ADDR_ERROR
03H/ 3	Destination address is not on a correct boundary.	DST_ADDR_ERROR
04H/ 4	Source address is not mapped in the memory map. Count value is taken in to consideration where applicable.	SRC_ADDR_NOT_MAPPED
05H/ 5	Destination address is not mapped in the memory map. Count value is taken in to consideration where applicable.	DST_ADDR_NOT_MAPPED
06H/ 6	Byte count is not multiple of 4 or is not a permitted value.	COUNT_ERROR
07H/ 7	Sector number is invalid or end sector number is greater than start sector number.	INVALID_SECTOR
08H/ 8	Sector is not blank.	SECTOR_NOT_BLANK
09H/ 9	Command to prepare sector for write operation was not executed.	SECTOR_NOT_PREPARED_FOR_WRITE_ OPERATION
0AH/ 10	Source and destination data not equal.	COMPARE_ERROR
0BH/ 11	Flash programming hardware interface is busy	BUSY
0CH/ 12	Insufficient number of parameters or invalid parameter.	PARAM_ERROR
0DH/ 13	Address is not on word boundary.	ADDR_ERROR
0EH/ 14	Address is not mapped in the memory map. Count value is taken in to consideration where applicable.	ADDR_NOT_MAPPED
0FH/ 15	Command is locked.	CMD_LOCKED
10H/ 16	Unlock code is invalid.	INVALID_CODE
11H/ 17	Invalid baud rate setting.	INVALID_BAUD_RATE
12H/ 18	Invalid stop bit setting.	INVALID_STOP_BIT
13H/ 19	Code read protection enabled.	CODE_READ_PROTECTION_ENABLED

EXAMPLES

Example 1: Read PartID

Figures 85, 86 and 87 show sequence of command and read cycles used to obtain a PartID value from a LPC2000 device.

As described in Table 36, in Command1 a parallel programmer sends 4 bytes starting with 0x36 to the LPC2000 device. Command2 sends 4 bytes too, while in Read1 cycle programmer reads 8 bytes from the device. Valid PartID is in the last 4 bytes of this group, and its value in this case is 0xFFF0 FF32, since the LPC2106 device was used.

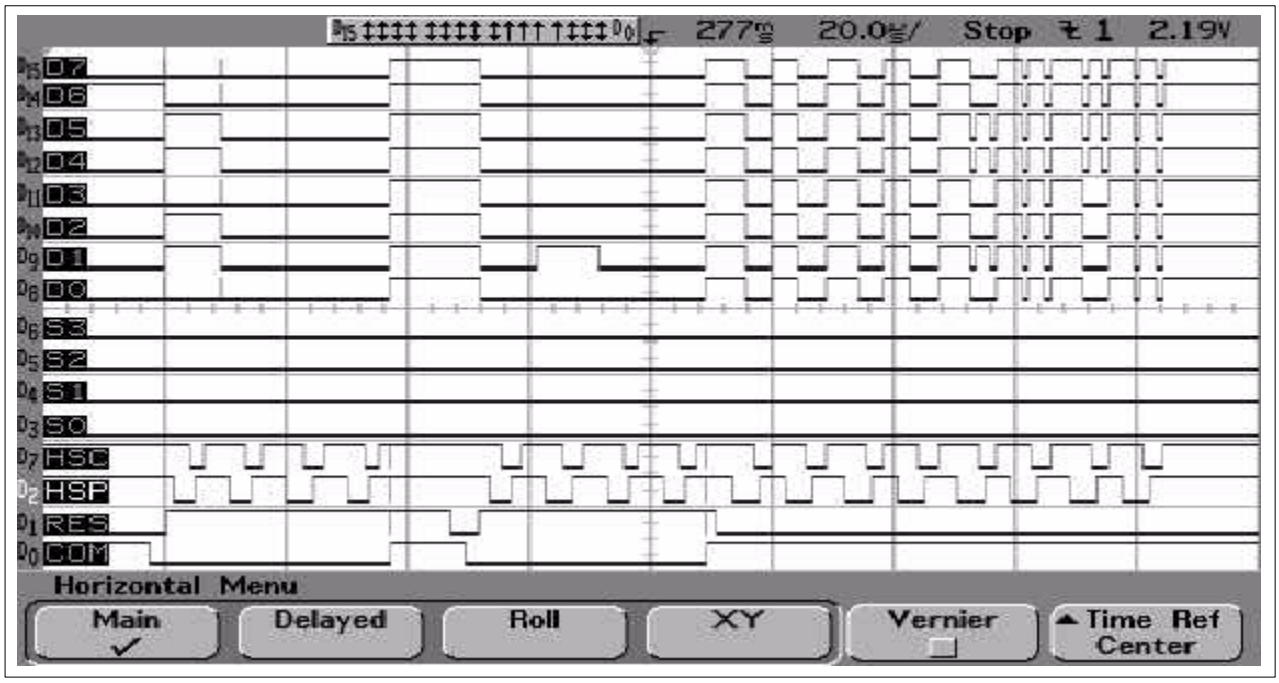


Figure 85: Read PartID Command1 (0x36...), Command2 (0x00, 0x02,...)and Read1 cycles (example shows PartID for LPC2106: 0xFFF0FF32)

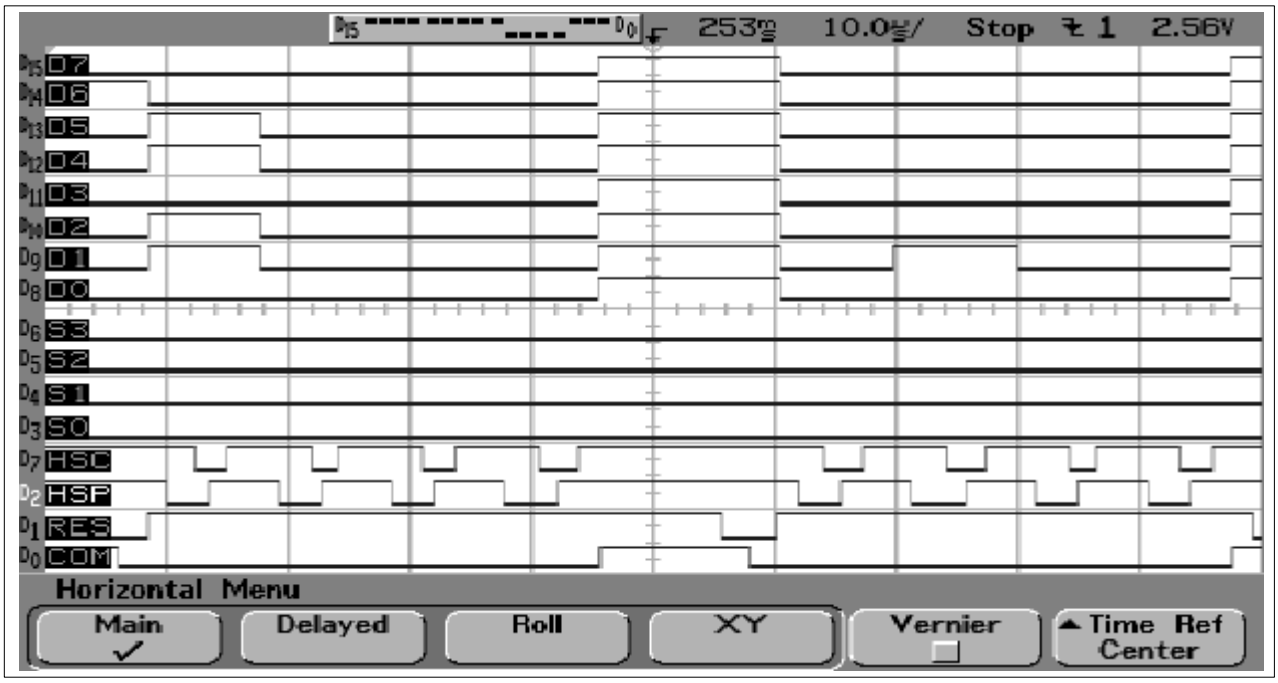


Figure 86: Read PartID Command1 (0x36...) and Command2 (0x00, 0x02,...) cycles

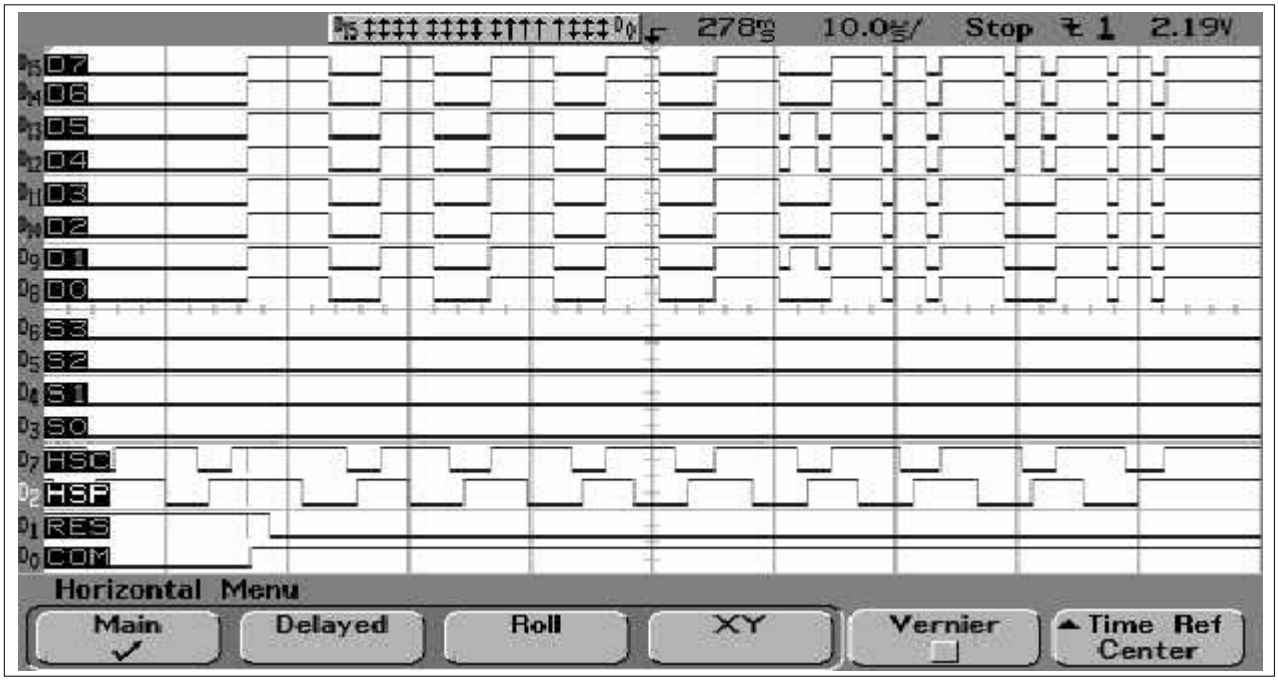


Figure 87: Read PartID Read1 cycle (example shows PartID for LPC2106: 0xFFF0FF32)

Example 2: Read Boot Code Version

Figures 88, 89 and 90 show sequence of command and read cycles used to obtain a Boot Code VersionI and a Boot Code VersionP values from the selected LPC2000 device.

As described in Table 36, a parallel programmer sends 4 bytes in Command1 starting with 0x37 to the LPC2000 device. Command2 sends 4 bytes too, while in Read1 cycle a parallel programmer reads 12 bytes from the programmed device. Valid Boot Code VersionI is contained in the 4 central bytes of the received answer, while Boot Code VersionP is represented in the last 4 bytes of the read-out. Example presented here corresponds to Boot Code VersionP=1.5 and Boot Code VersionI=1.0.

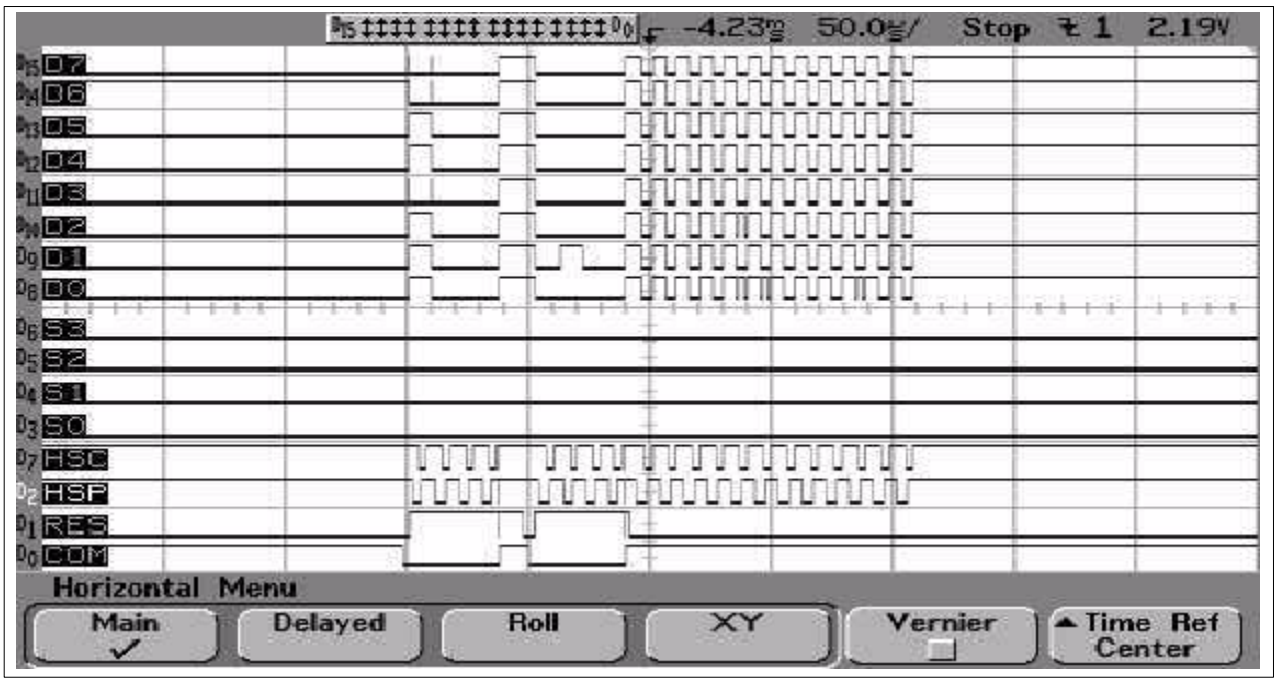


Figure 88: Read Boot Code Version Command1 (0x37...), Command2 (0x00, 0x02,...) and Read1 cycles (example shows CodeVersionI=0x0000 0015 and CodeVersionP=0x0000 0010)

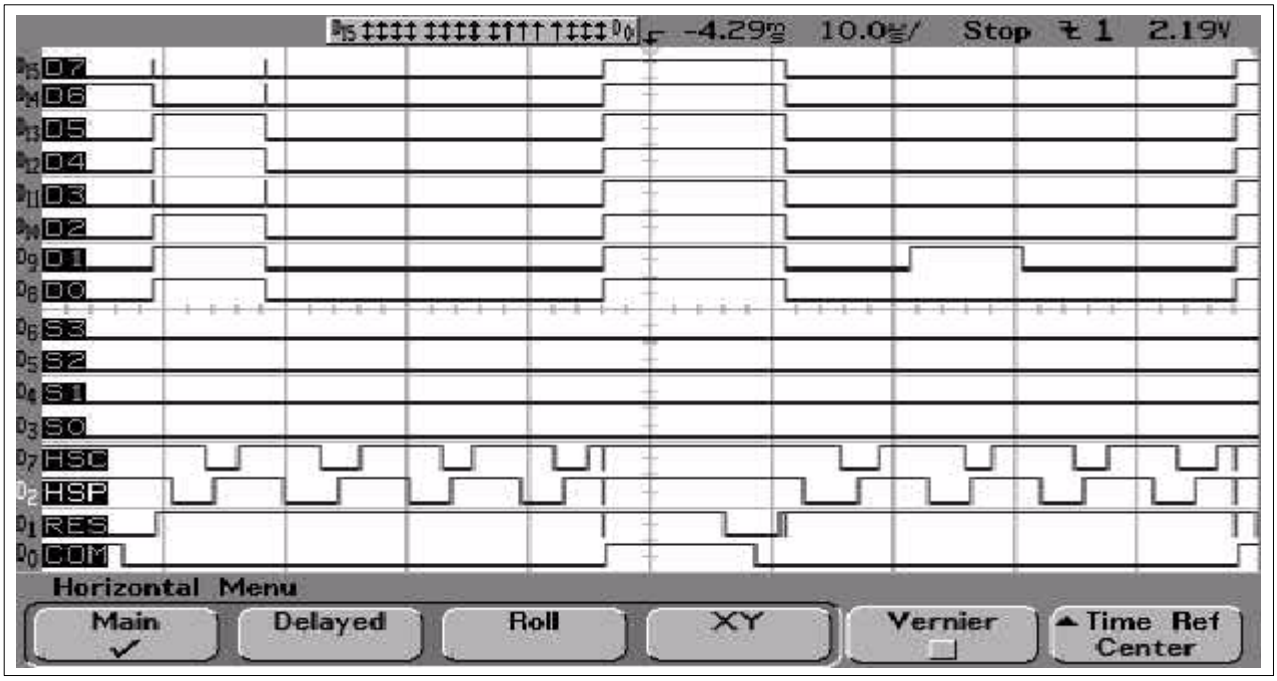


Figure 89: Read Boot Code Version Command1 (0x37...) and Command2 (0x00, 0x02,...) cycles

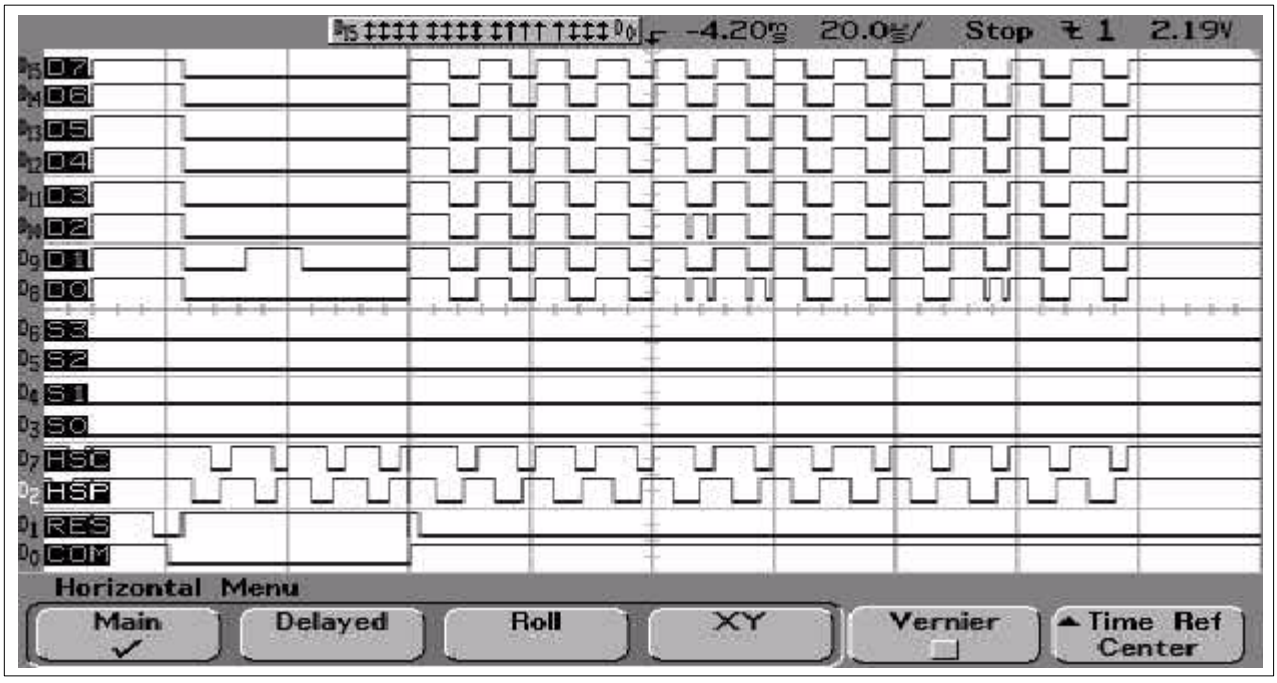


Figure 90: Read Boot Code Version Command2 (0x00, 0x02, ...) and Read1 cycles (example shows CodeVersionI=0x0000 0015 and CodeVersionP=0x0000 0010)

Example 3: Erase Flash Memory

Figures 91 and 92 show sequence of command cycles used to erase sector 1 of a LPC2000 Flash memory.

Figure 91 illustrates 400 millisecond time interval required for any number of Flash memory sectors to be erased: erasing of a single sector takes the same amount as performing the full chip erase.

In this particular example, only sector 1 (address range 0x0000 2000 to 0x0000 3FFF) was erased.

Command1 sent 12 bytes starting with 0x32. Command2 sent 16 bytes, starting with 0x34.

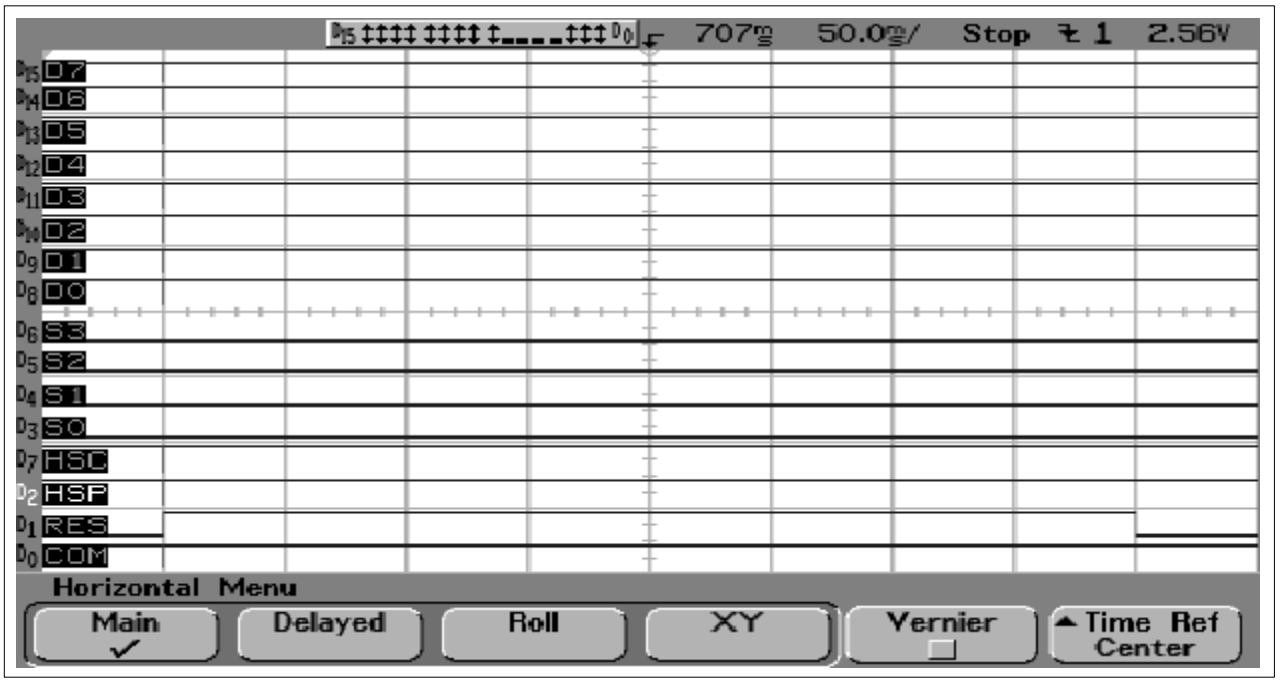


Figure 91: Erase Flash Memory procedure lasts at least 400 ms
(RES line stays high for 400 ms which indicates the duration of the erase cycle)

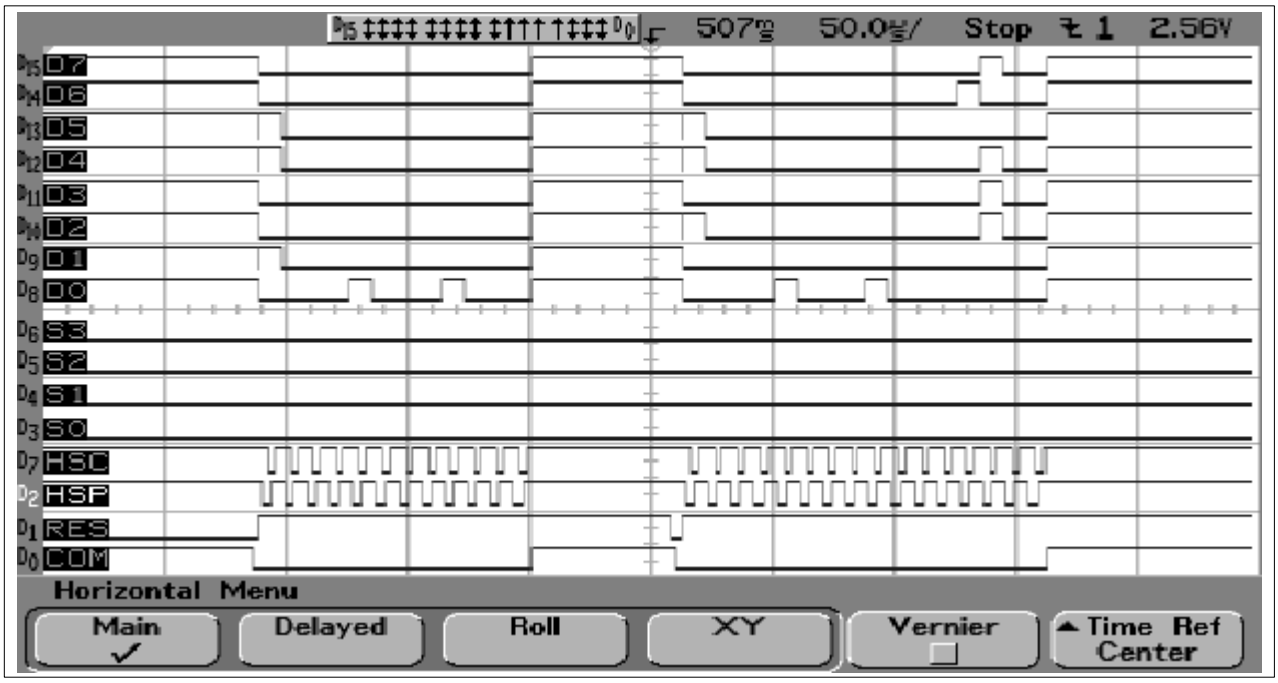


Figure 92: Erase Flash Memory Command1 (0x32, ...) and Command2 (0x34, ...) cycles

Example 4: Program Flash Memory and Verify

Note: *this example was written for the parallel interface 1.0 (CodeVersionP = 0x0000 0010) when up to 8kB of data could be programmed into the Flash at once. As of parallel interface revision 2.0 (the one covered with this specification), the amount of transferred data is limited to a 1kB. As a result, certain changes in parameters had to be made, and consequently figures that follow do not match bit for bit what is described in the previous text. However, the applied concept of data loading and programming has not changed over the time.*

Figures 93 to 97 show sequence of command cycles used to program 8 kB of data into the Flash memory sector 1 of the LPC2106 microcontroller.

While data transfer rate between a parallel programmer and a programmed device depends on the throughput of the parallel interface, programming itself requires fixed amount of time and for 8 kB size it is close to 35 ms (Figure 95).

Figure 94 starts with a parallel programmer sending Command1. This 0x32 leads stream of 12 bytes that specify Flash sector 1 for write operation. Command2 follows and its first four bytes specify 0x33 followed by three 0x00 bytes. The next 4 bytes contain FlashAddr = 0x000 2000. Third set of four bytes specify system constant (0x4000 1014), and they are followed by the number of bytes that will be written into the Flash memory - 8192 (0x0000 2000 again). Another system constant follows (0x0000 9C40).

After this point, block of 8192 bytes is sent over the parallel interface, staring with 0x00, 0x01, 0x02...

Figure 96 shows Command3 cycle when the LPC2106 was instructed to compare content of the Flash memory from 0x0000 2000 and the RAM buffer residing on a fixed location. Number of bytes for comparing is 8192.

At the end, Figure 97 illustrates the time necessary for these data to be compared and status lines to become updated. For 8 kB block, some 5.5 ms are required.

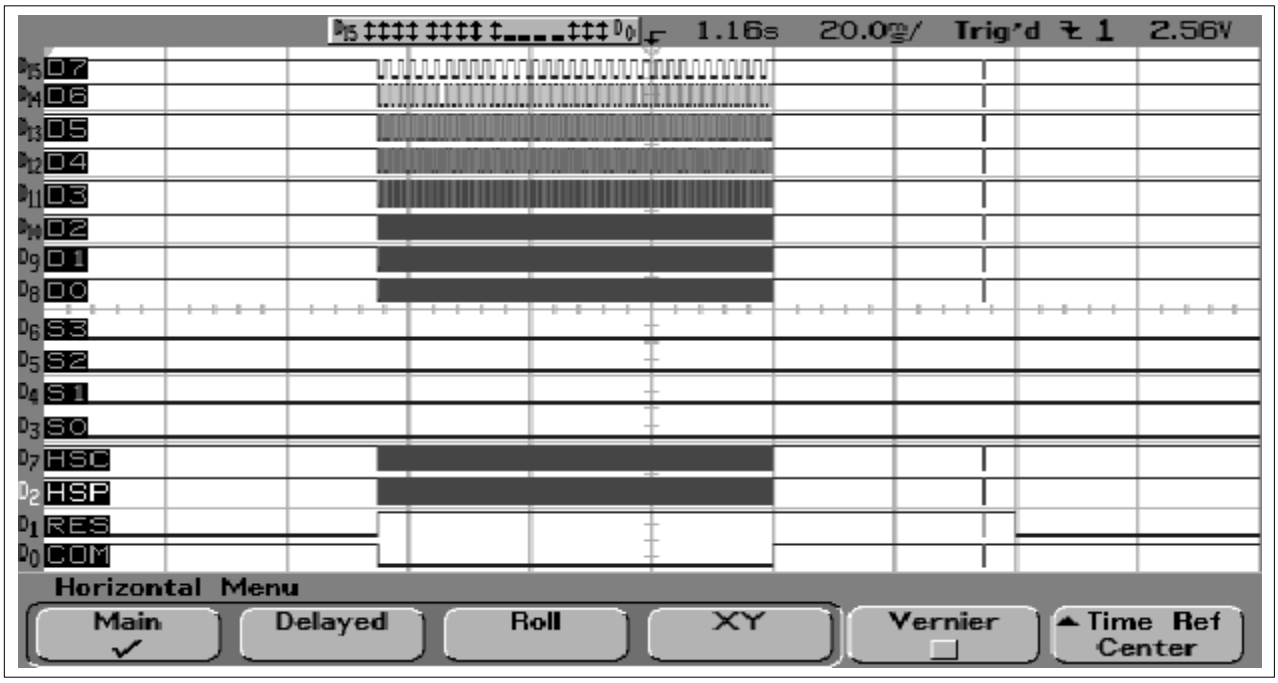


Figure 93: Program Flash Memory and Verify - Command1, Command2 and Command3 cycles

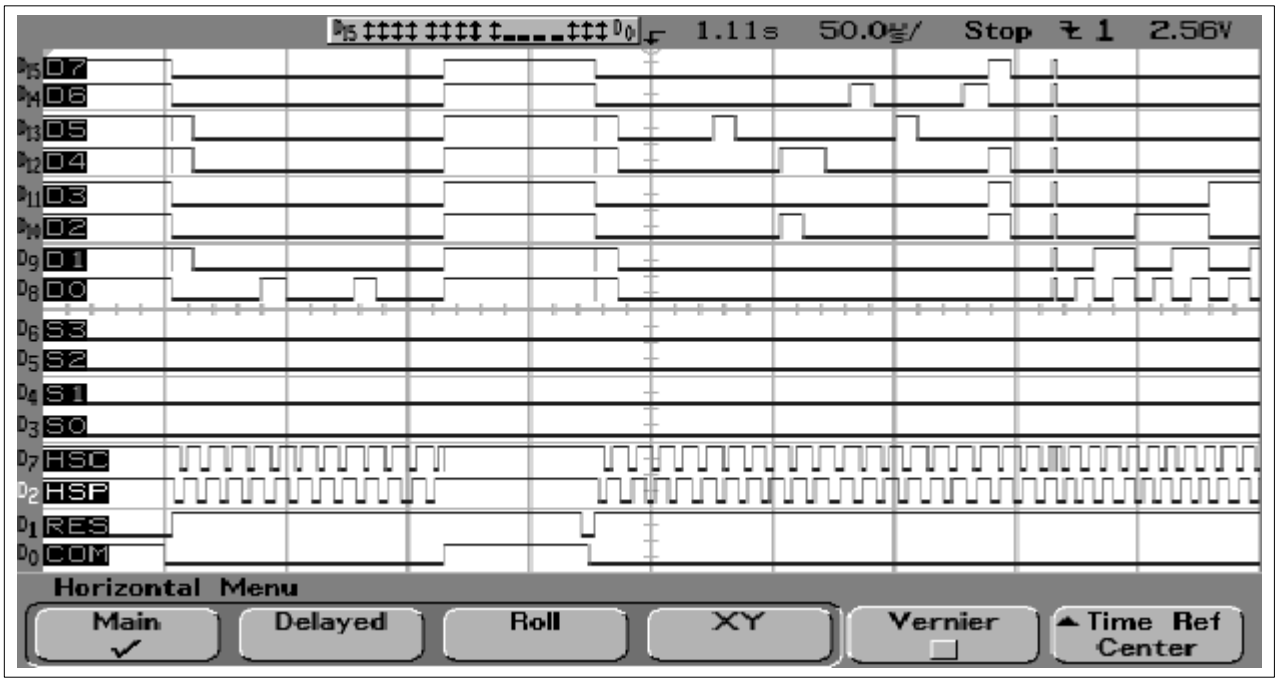


Figure 94: Program Flash Memory and Verify - Command1 (0x32, ...) and the beginning of Command2 (0x33, ...) cycle

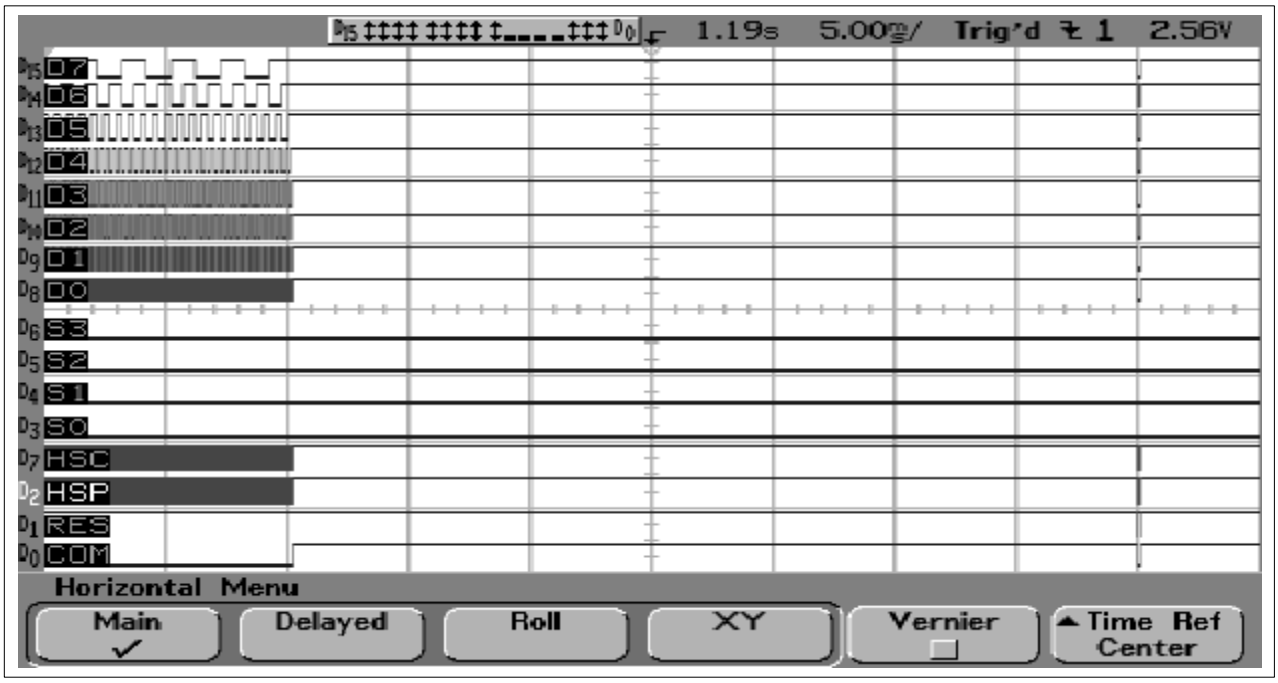


Figure 95: Program Flash Memory and Verify - Command2 cycle end and the delay before Command3 (example shows ~35 ms required for 8 kB to be programmed - delay between edges on COM and RES)

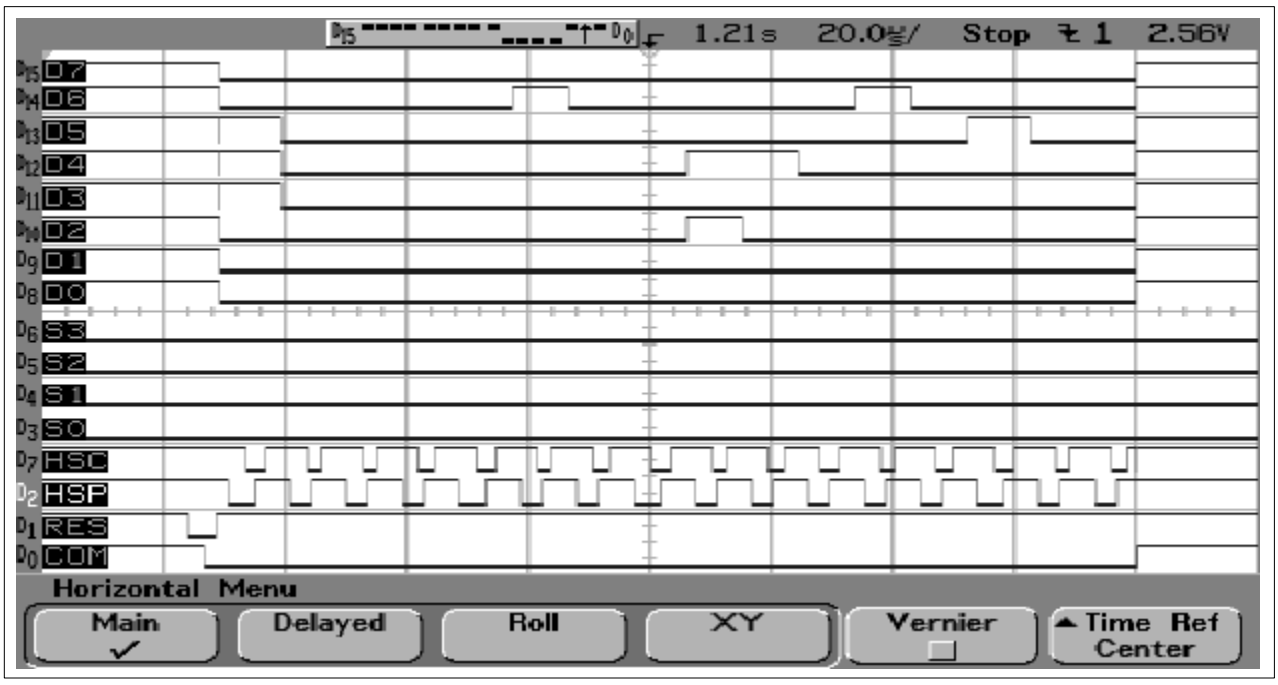


Figure 96: Program Flash Memory and Verify - Command3 (0x38, ...) cycle

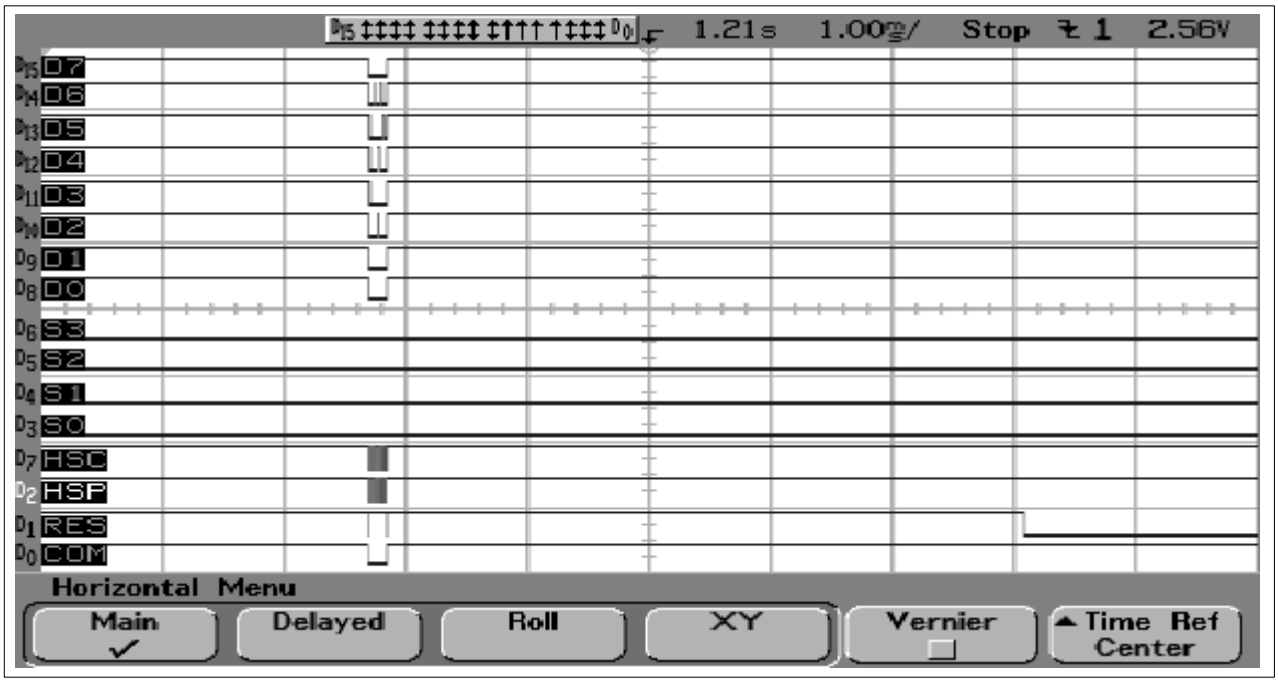


Figure 97: Program Flash Memory and Verify - Command2 and Command3 cycles and the final status update (example shows ~5.4 ms for 8 kB to be verified - delay between edges on COM and RES lines)