The function of AOI and crossbar modules

The AOI and crossbar switch modules are integrated for MC56F84xxx and MC56F82xxx DSC family, KV4x and KV5x Kinetis KV family and i.mxRT family. The KV58 has two Crossbar switches, one is crossbar switch A, another is crossbar switch B.

As you know the crossbar switch has input signals source and output signal source, the function of crossbar switch is to route the input signals to the output signal source

For KV58, the Crossbar switch A can route the triggering signal sources to the triggering signal of on-chip peripherals or chip pads. The crossbar switch B can route the signal source to AOI input. The target of the AOI and Crossbar switch B modules is to generate complicated triggering signal for on-chip peripherals for example ADC, DAC….

The AOI module can implement AND/OR/INVERT operation for both internal and external signals, for DSC and KV families, the AOI module input signals are from Crossbar Switch B, it's 4 output signal are routed to Crossbar Switch A.

The DOC is only based on KV58 family, but it's content adapts to DSC family, KV family and i.mxrt family.

1. Crossbar switch B introduction

1.1.CrossbarB input introduction

The crossbar switch B is a network, it can route the input signals to the input pins of AOI module, furthermore, one input signal can route to multiple input signal of AOI module, but one AOI input signal can be connected to ONLY ONE input signal.

The input/output signals are chip specific, user has to check the reference manual or data sheet for the information:

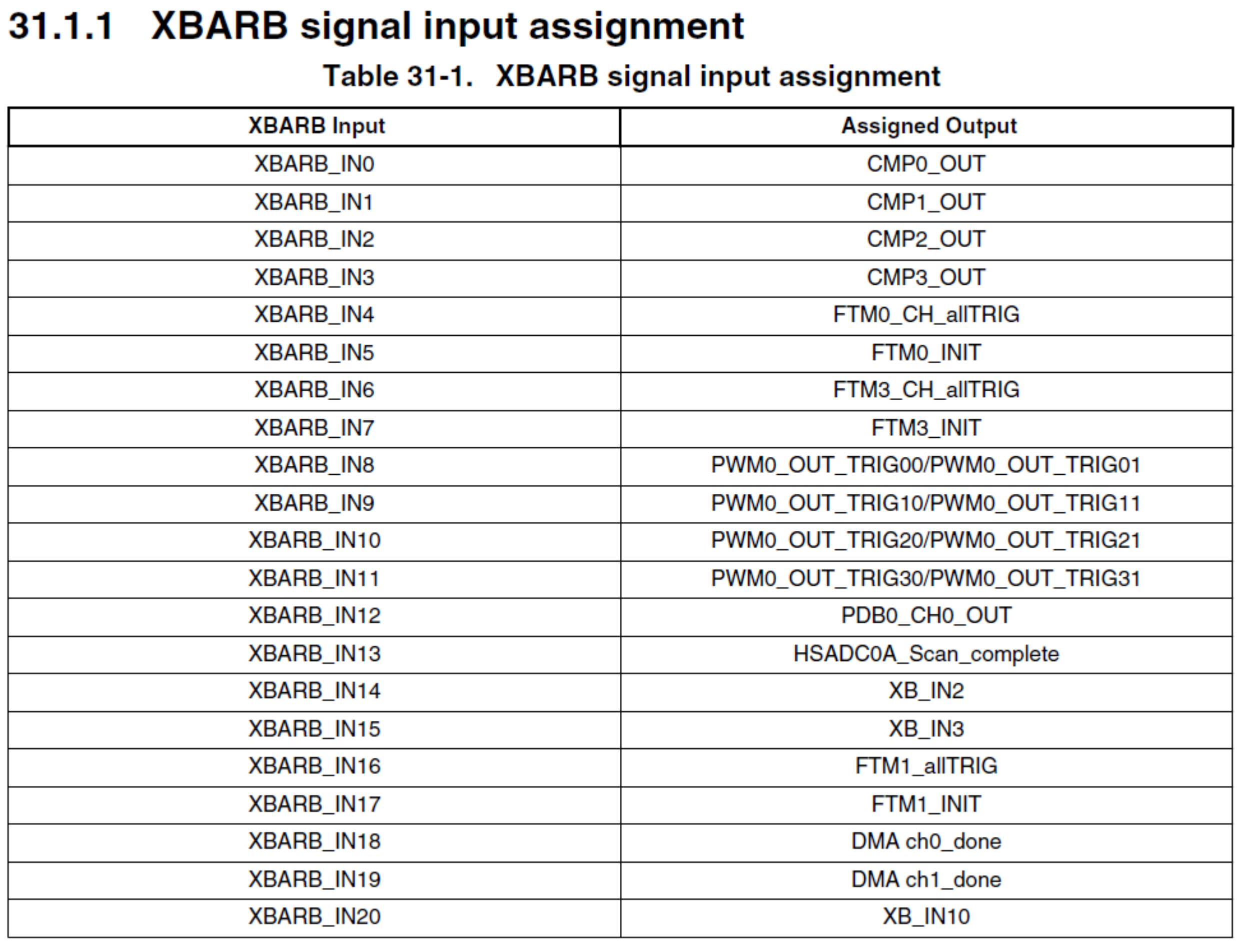
This is the input signals of AOI for KV58. The AOI module has 16 input signal nodes AOI_IN0~AOI_IN15, which are connected directly to the 16 output signals of Corssbar switch B (XBARB). The AOI module has 4 sub-modules, each AOI sub-module has 4 input pins. For example, AOI_IN0/1/2/3 belongs to AOI sub-module0, AOI_IN4/5/6/7 belongs to AOI sub-module1…

In the following Table 31-2, the crossbar switch B output0 XBARB_OUT0 is connected to AOI_IN0, the crossbar switch B output1 XBARB_OUT1 is connected to AOI_IN1…, there are 16 output nodes for crossbar switch B, or 16 input node for AOI module.

In the following Table 31-1, the actual signals are listed, the signals on the "Assigned Output" column are output signals of on-chip peripherals or external input pads, they can be routed to the input nodes of AOI module and do the AND/OR/Invert operation. You can select ONE signal from the table 31-1 and route it to any one of XBARB_OUTx, in other words to AOI_INx pins. For example, if user wants to AND two external signals, which are connected to pads XB_IN2 and XB_IN3 pads, user can use any AOI sub-module to do it, here I select to use the AOI sub-module 0 to do the AND operation, the sub-module0 of AOI has 4 input signals: AOI_IN0, AOI_IN1, AOI_IN2 and AOI_IN3, the XBARB_OUT0 is connected to AOI_IN0, XBARB_OUT1 is connected to AOI_IN1, the XBARB_OUT0 signal corresponds to the Low byte of XBARB_SEL0, the XBARB_OUT1 signal corresponds to the high byte of XBARB_SEL0. The XB_IN2 and XB_IN3 signals correspond to input channel 14 and 15 based on Table 31-1, so it is okay to configure the XBARB_SEL0:

```
XBARB_SEL0=(15<<8)|14;
```

With the above code, the external signals from pads XB_IN2/XB_IN3 are routed to the input of sub-module0 of AOI.



31.1.1 XBARB signal input assignment

Table 31-1. XBARB signal input assignment

XBARB Input	Assigned Output
XBARB_IN0	CMP0_OUT
XBARB_IN1	CMP1_OUT
XBARB_IN2	CMP2_OUT
XBARB_IN3	CMP3_OUT
XBARB_IN4	FTM0_CH_allTRIG
XBARB_IN5	FTM0_INIT
XBARB_IN6	FTM3_CH_allTRIG
XBARB_IN7	FTM3_INIT
XBARB_IN8	PWM0_OUT_TRIG00/PWM0_OUT_TRIG01
XBARB_IN9	PWM0_OUT_TRIG10/PWM0_OUT_TRIG11
XBARB_IN10	PWM0_OUT_TRIG20/PWM0_OUT_TRIG21
XBARB_IN11	PWM0_OUT_TRIG30/PWM0_OUT_TRIG31
XBARB_IN12	PDB0_CH0_OUT
XBARB_IN13	HSADC0A_Scan_complete
XBARB_IN14	XB_IN2
XBARB_IN15	XB_IN3
XBARB_IN16	FTM1_allTRIG
XBARB_IN17	FTM1_INIT
XBARB_IN18	DMA ch0_done
XBARB_IN19	DMA ch1_done
XBARB_IN20	XB_IN10

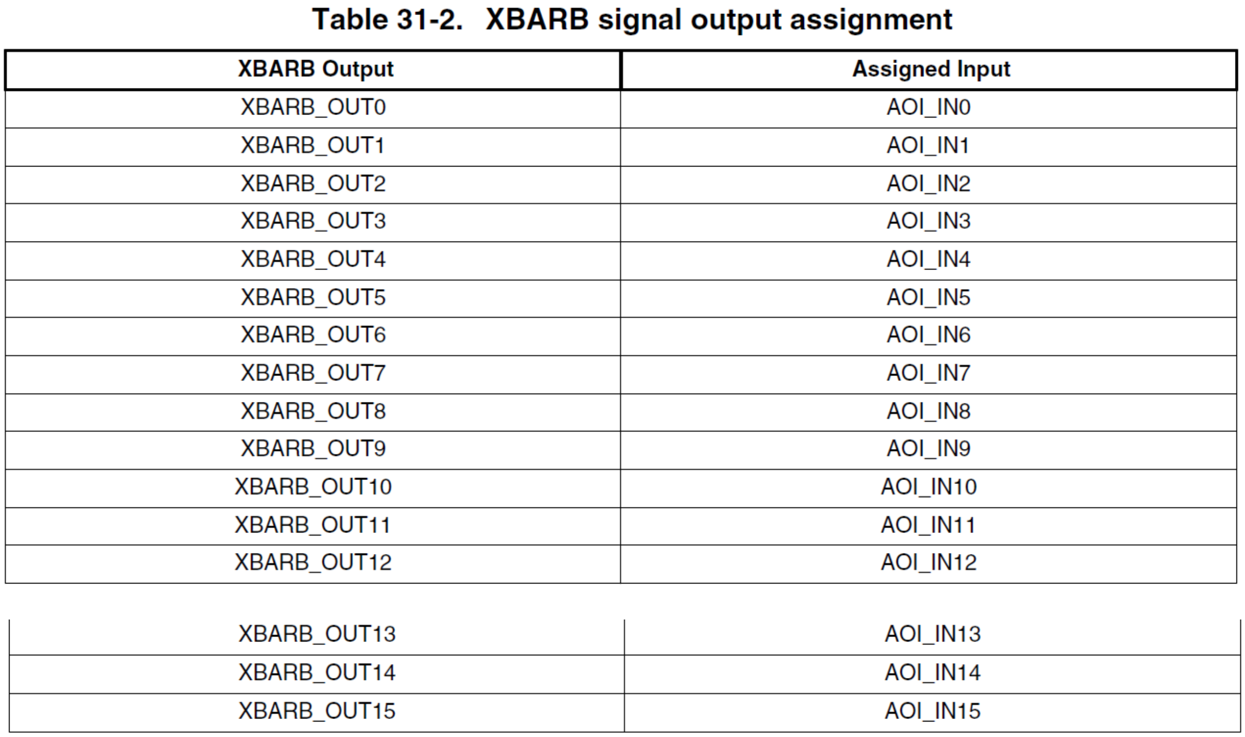


Table 31-2. XBARB signal output assignment

XBARB Output	Assigned Input
XBARB_OUT0	AOI_IN0
XBARB_OUT1	AOI_IN1
XBARB_OUT2	AOI_IN2
XBARB_OUT3	AOI_IN3
XBARB_OUT4	AOI_IN4
XBARB_OUT5	AOI_IN5
XBARB_OUT6	AOI_IN6
XBARB_OUT7	AOI_IN7
XBARB_OUT8	AOI_IN8
XBARB_OUT9	AOI_IN9
XBARB_OUT10	AOI_IN10
XBARB_OUT11	AOI_IN11
XBARB_OUT12	AOI_IN12
XBARB_OUT13	AOI_IN13
XBARB_OUT14	AOI_IN14
XBARB_OUT15	AOI_IN15

1.2 **crossbar switch A introduction**

As you know the crossbar switch has input signals source and output signal source, the function of crossbar switch is to route the input signals to the output signal source, the input signal sources of crossbar switch A are on-chip peripheral output signals, the output signal sources of crossbar switch A are on-chip peripheral triggering input signals or chip pads, the chip pads can output signal.

The Assigned Output column of Table 30-1 XBARA input assignment lists the input signals of crossbar switch A, all of them are either on-chip peripheral output signals or external pads.

The Assigned Input column of Table 30-2 XBARA Output Signal assignment lists the output signals of crossbar switch A, all of them are either on-chip peripheral triggering input signals or chip pads.

1.3 **AOI outputs routing**

The AOI module has 4 sub-modules, each AOI sub-module has only ONE output signal called Event n, the "n" is from 0 to 3. The 4 AOI sub-module output signal Event n outputs are routed to crossbar switch A input nodes.

From the Table 30-1. XBARA input assignment, the AOI sub-module0 output signal Event0 is AND_OR_INVERT_0, the AOI sub-module1 output signal Event1 is AND_OR_INVERT_1, the AOI sub-module2 output signal Event2 is AND_OR_INVERT_2, the AOI sub-module3 output signal Event3 AND_OR_INVERT_3.

All the 4 AOI sub-module output signals Event n are the input sources of crossbar switch A, can be routed to crossbar switch A output signals, which are actual triggering signals of on-chip peripherals.

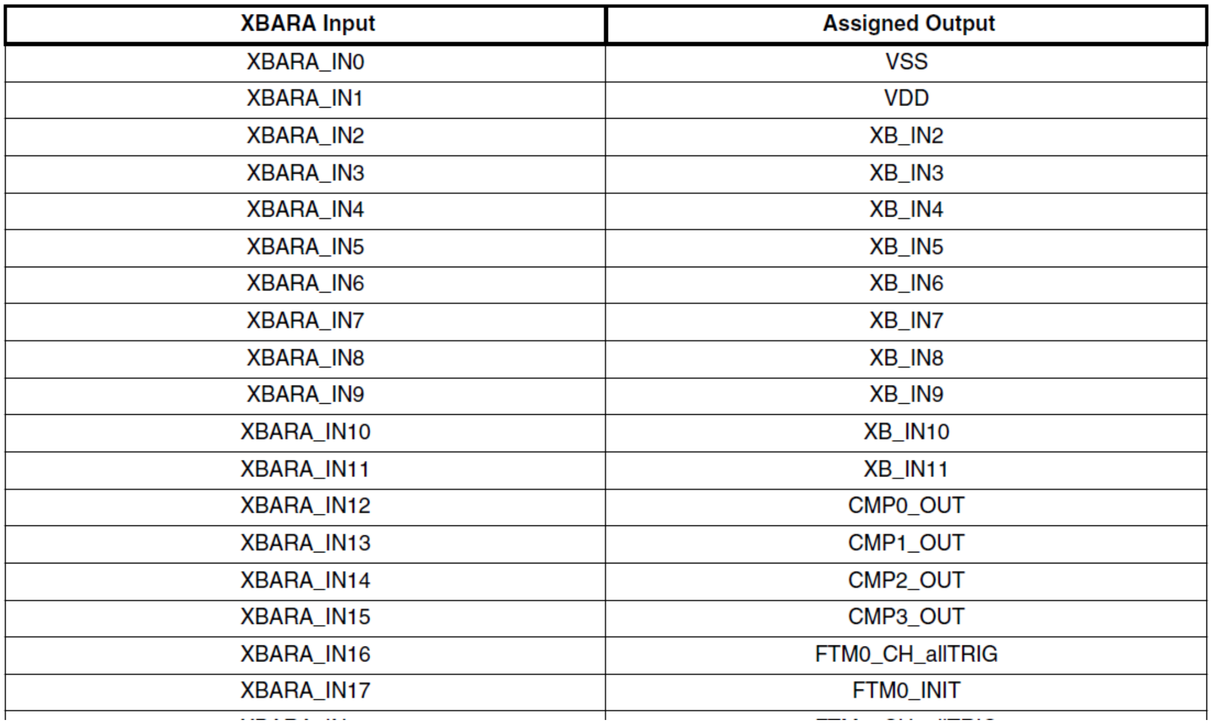
For example, the AOI sub-module 0 output signal Event 0(AND_OR_INVERT_0) is index 46 of crossbar switch A inputs called XBARA_IN46 as the following Table 30-1 XBARA input assignment. The

The XB_OUT7 signal is the index 7 of crossbar A output called XBARA_OUT7, as the following Table 30-2 XBARA signal Output assignment. Each two consecutive output signals of crossbar switch A corresponds ONE register XBARA_SELx.

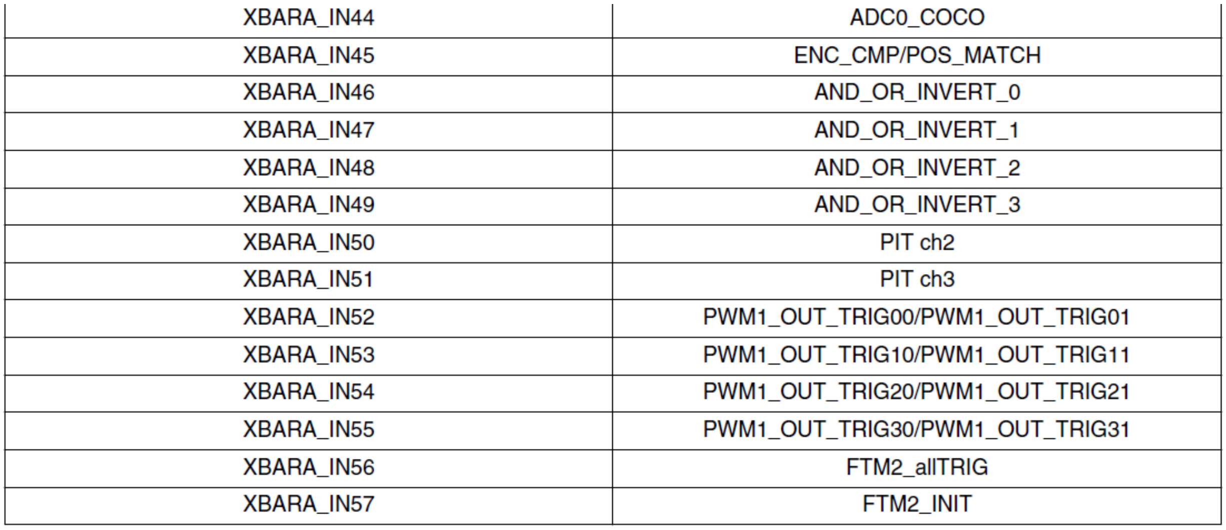
If you want to route the AOI sub-module 0 output signal Event 0 signal AND_OR_INVERT_0 to XB_OUT7, the XBARA_OUT7 corresponds to High byte of XBARA_SEL3, it is okay to use the code

XBARA_SEL3=46<<8;

Table 30-1. XBARA input assignment

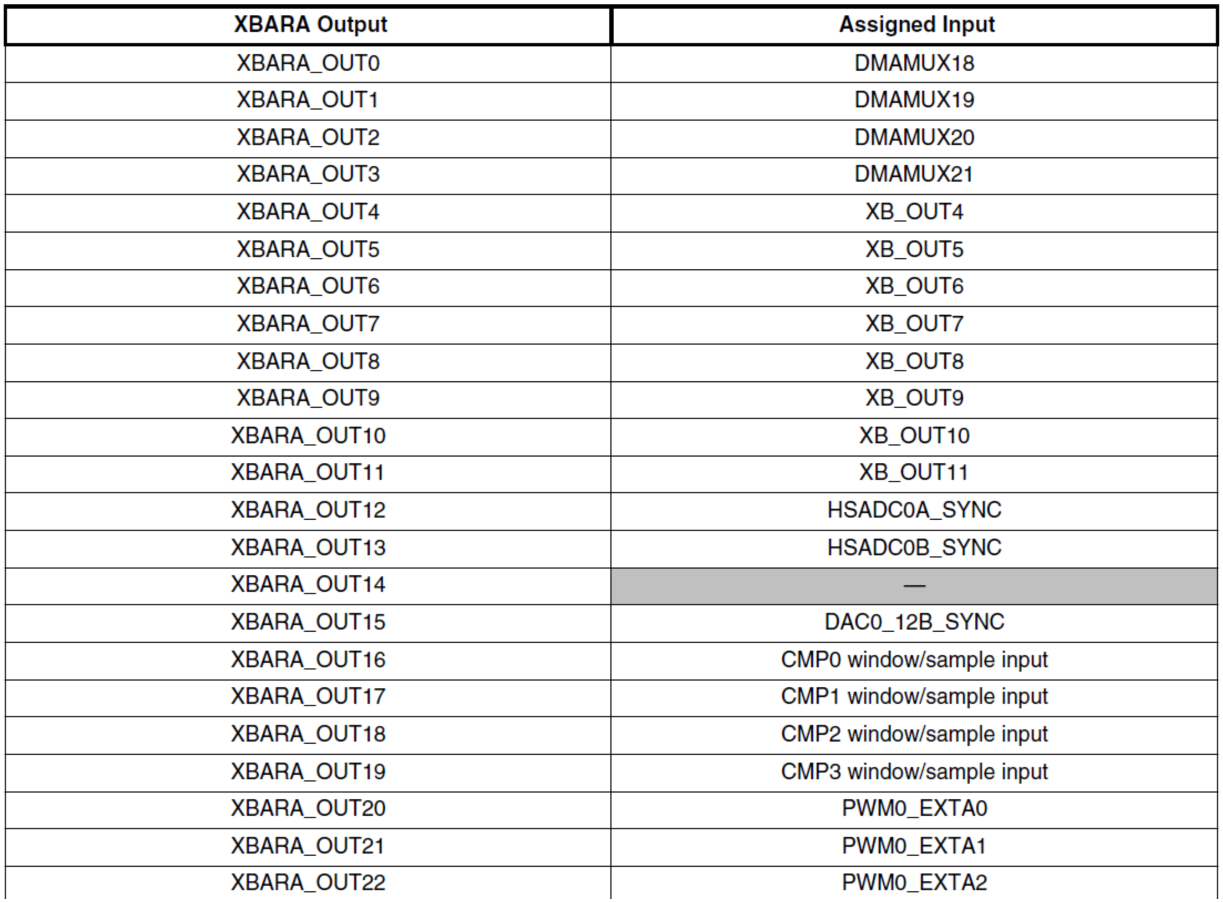


XBARA Input	Assigned Output
XBARA_IN0	VSS
XBARA_IN1	VDD
XBARA_IN2	XB_IN2
XBARA_IN3	XB_IN3
XBARA_IN4	XB_IN4
XBARA_IN5	XB_IN5
XBARA_IN6	XB_IN6
XBARA_IN7	XB_IN7
XBARA_IN8	XB_IN8
XBARA_IN9	XB_IN9
XBARA_IN10	XB_IN10
XBARA_IN11	XB_IN11
XBARA_IN12	CMP0_OUT
XBARA_IN13	CMP1_OUT
XBARA_IN14	CMP2_OUT
XBARA_IN15	CMP3_OUT
XBARA_IN16	FTM0_CH_allTRIG
XBARA_IN17	FTM0_INIT



XBARA_IN44	ADC0_COCO
XBARA_IN45	ENC_CMP/POS_MATCH
XBARA_IN46	AND_OR_INVERT_0
XBARA_IN47	AND_OR_INVERT_1
XBARA_IN48	AND_OR_INVERT_2
XBARA_IN49	AND_OR_INVERT_3
XBARA_IN50	PIT ch2
XBARA_IN51	PIT ch3
XBARA_IN52	PWM1_OUT_TRIG00/PWM1_OUT_TRIG01
XBARA_IN53	PWM1_OUT_TRIG10/PWM1_OUT_TRIG11
XBARA_IN54	PWM1_OUT_TRIG20/PWM1_OUT_TRIG21
XBARA_IN55	PWM1_OUT_TRIG30/PWM1_OUT_TRIG31
XBARA_IN56	FTM2_allTRIG
XBARA_IN57	FTM2_INIT

Table 30-2. XBARA signal output assignment



XBARA Output	Assigned Input
XBARA_OUT0	DMAMUX18
XBARA_OUT1	DMAMUX19
XBARA_OUT2	DMAMUX20
XBARA_OUT3	DMAMUX21
XBARA_OUT4	XB_OUT4
XBARA_OUT5	XB_OUT5
XBARA_OUT6	XB_OUT6
XBARA_OUT7	XB_OUT7
XBARA_OUT8	XB_OUT8
XBARA_OUT9	XB_OUT9
XBARA_OUT10	XB_OUT10
XBARA_OUT11	XB_OUT11
XBARA_OUT12	HSADC0A_SYNC
XBARA_OUT13	HSADC0B_SYNC
XBARA_OUT14	—
XBARA_OUT15	DAC0_12B_SYNC
XBARA_OUT16	CMP0 window/sample input
XBARA_OUT17	CMP1 window/sample input
XBARA_OUT18	CMP2 window/sample input
XBARA_OUT19	CMP3 window/sample input
XBARA_OUT20	PWM0_EXTA0
XBARA_OUT21	PWM0_EXTA1
XBARA_OUT22	PWM0_EXTA2

2). **AOI module**

The AOI is the acronym of AND/OR/INVERT, in other words, it can do the combination operation of AND, OR and Invert. There are 4 sub-modules for the AOI module: sub-module0, sub-module1, sub-module2 and sub-module3. Each sub-module has 4 input signal: AOI_IN[4*n], AOI_IN[4*n+1], AOI_IN[4*n+2] and AOI_IN[4*n+3]. For example, for the sub-module1, the “n” is 1, the four inputs are AOI_IN4, AOI_IN5, AOI_IN6, AOI_IN7.

The following Figure 29-1 Simplified AOI Block Diagram is the logic diagram of one of four AOI sub-modules, each AOI submodule has 4 input signals: An/Bn/Cn/Dn and one output signal: Eventn. From the AOI block diagram, we see that there are 4 AND gates whose inputs can be configured by firmware, A pair of 16-bit registers configures this four AND gates of AOI function, the BFCRT01n register defines the configuration of AND gate 0 and 1, and the BFCRT23n register defines the configuration of AND gate 2 and 3.

The AOI module provides a universal Boolean function generator using a four-term sum of products expression with each product term containing true or complement values of the four selected event inputs (An, Bn, Cn, Dn). Specifically, the EVENTn output is defined by the following "4 x 4" Boolean expression:

```
EVENTn
= (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 0
| (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 1
| (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 2
| (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 3
```

Each multiplexer includes 4 inputs, they are logic 0, logic 1, input signal itself and inverter of the input signal, but the Figuer 29-1 only list 2 terms(input signal itself and inverter of the input signal), it misses the fixed logic 0 and logic 1.

The section **Boolean Function Term 0 and 1 Configuration Register for EVENTn** (AOIx_BFCRT01n) represents the AOI register, each AOI sub-module is controlled by a pair of register, for example sub-module0 is controlled by AOIx_BFCRT010 and AOIx_BFCRT230. Each AND gate includes 4 inputs signals, which are outputs of 4 multiplexer, each multiplexer can select logic 0, logic1, input signal itself and inverter of the input signal, which one is selected for the multiplexer is controlled by the two bits PTxx.

29.4.1 Boolean Function Term 0 and 1 Configuration Register for EVENTn (AOIx_BFCRT01*n*)

Address: 4005_B000h base + 0h offset + (4d × i), where i=0d to 3d

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	PT0_AC		PT0_BC		PT0_CC		PT0_DC		PT1_AC		PT1_BC		PT1_CC		PT1_DC	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AOIx_BFCRT01*n* field descriptions

Field	Description
15–14 PT0_AC	Product term 0, A input configuration This 2-bit field defines the Boolean evaluation associated with the selected input A in product term 0. 00 Force the A input in this product term to a logical zero 01 Pass the A input in this product term 10 Complement the A input in this product term 11 Force the A input in this product term to a logical one

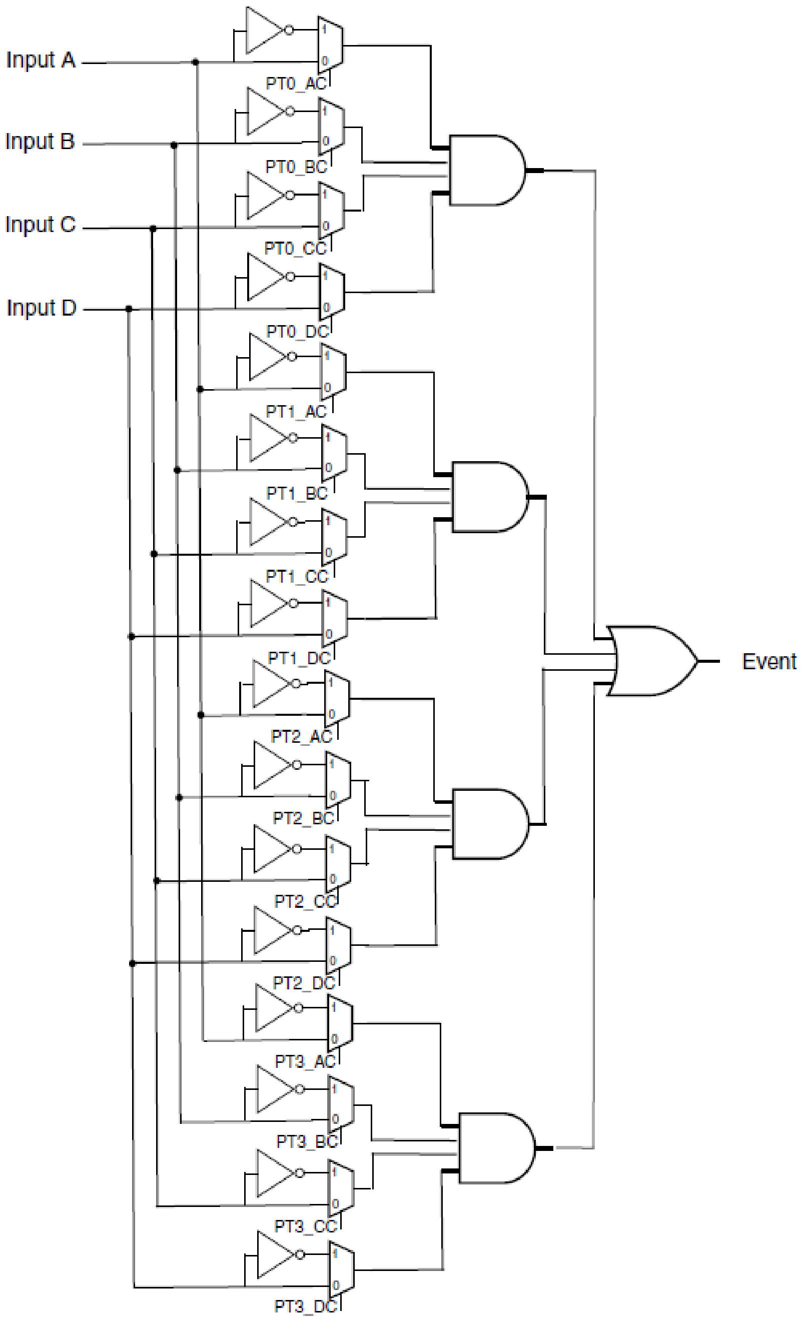
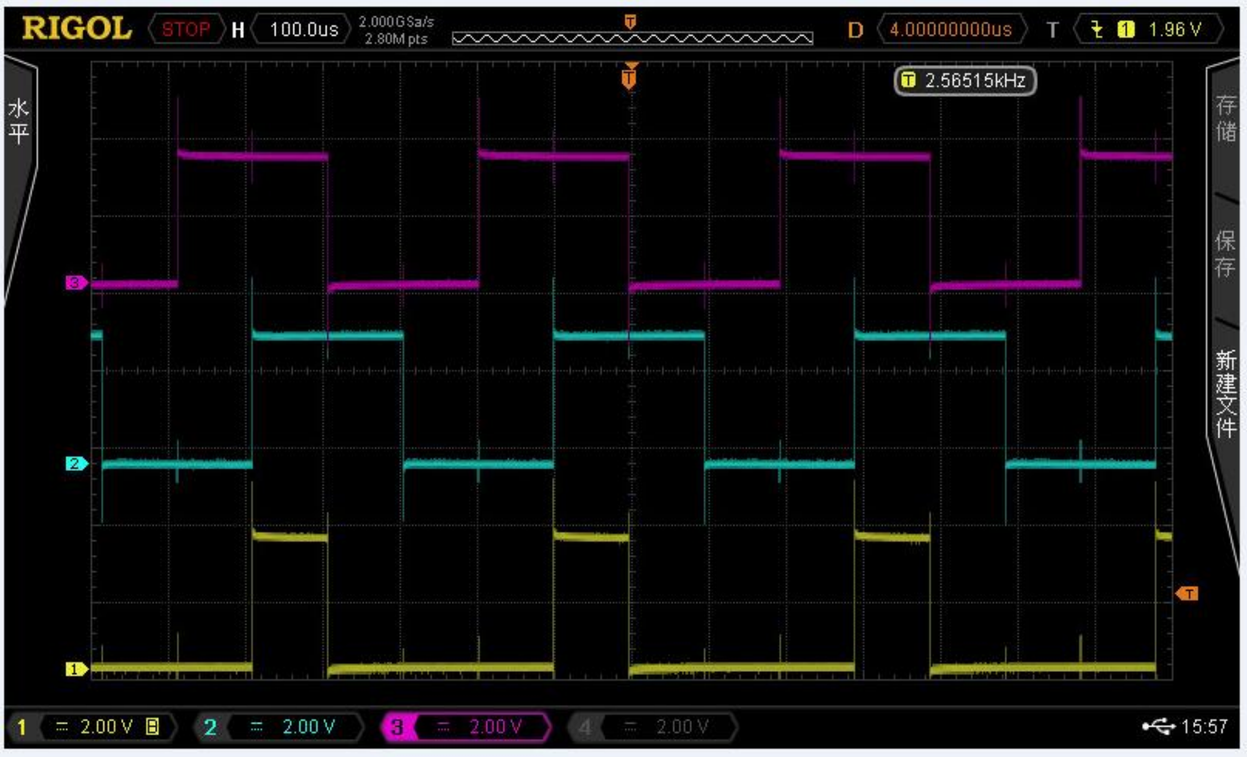


Figure 29-1. Simplified AOI Block Diagram

3)**Test and conclusion**

Based on KDS ver3.2 tools and TWR-KV58F220, the example demos how to implement AND operation of two signals via crossbar switch A and B and AOI modules.

The two logic signals are connected to the pads of KV58, and routed to AOI inputs via Crossbar switch B, the AOI sub-module0 implements the AND operation of the two signals, and output the AND output signal Event0 to pad of KV58 via crossbar switch A. Connect input pads and output pad of KV58 to oscilloscope, from the waveform of the three signals on scope, we can see that the AND logic is implemented.



On the screenshot of oscilloscope, the channel 3 is input signal from XB_IN2 (PTC5, A70 of primary board)

channel 2 is input signal from XB_IN3 (PTC6,A71 of primary Elevator board) on

channel 1 is output signal from XB_OUT7 (PTC7,A72 of primary Elevator board)

we can see that the channel 1 signal is the AND of channel 2 and channel 3 signals.

Appendix: code snippet.

The example code is developed under KDS 3.2 tools, TWR-KV58F220 board and Primary/Secondary Elevator board by X.J Rong(xiangjun.rong@nxp.com)

//The demo code demo how to set up the crossbar switch A and B and AOI module

//The demo code implement an AND circuit from external pins XB_IN2 and XB_IN3, and output the AND //signal to XB_OUT7 pad

```
void AOIpinAssignment(void)

{
        //configure the pins PTC5, PTC6 and PTC7 as XB_IN2, XB_IN3 and XB_OUT7
        //enable PTC gated clock
        SIM_SCGC5|=0x800; //setting PORTC bit
        PORTC_PCR5&=~(0xF00);
```

```c
	PORTC_PCR5|=4<<8;

	PORTC_PCR6&=~(0xF00);
	PORTC_PCR6|=4<<8;

	PORTC_PCR7&=~(0xF00);
	PORTC_PCR7|=6<<8;
}

//using AOI0 module
//XBARB_IN14 XB_IN2
//XBARB_IN15 XB_IN3

//XBARB_OUT0 AOI_IN0
//XBARB_OUT1 AOI_IN1

//XBARA_OUT7 XB_OUT7
//XBARA_IN46 AND_OR_INVERT_0
void crossbarSetting(void)
{
	//enable crossbar gated clock
	SIM_SCGC5|=0x6000000;
	XBARB_SEL0=(15<<8)|14;

	XBARA_SEL3|=46<<8;
}


void AOISetting(void)
{
	//enable AOI gated clock
	SIM_SCGC5|=0x8000000;
	AOI0_BFCRT010=0x5F00;
}
```