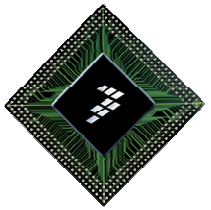
Sept, 2010

NVIC
Nested Vector Interrupt Controller



freescale™
semiconductor

NVIC Training Outline

1. Module Overview
2. On-chip interconnects and inter-module dependencies
3. Software configuration
4. Typical use cases
5. Demo code explanation
6. Frequently asked question list
7. Reference material

freescale™ semiconductor

NVIC Training Outline

1. **Module Overview**
2. On-chip interconnects and inter-module dependencies
3. Software configuration
4. Typical use cases
5. Demo code explanation
6. Frequently asked question list
7. Reference material

a) Block Diagram
b) Feature list
c) Key value add components

freescale™ semiconductor

NVIC Feature list

- Low latency Interrupt
  - 12 cycles to PUSH on ISR entry
  - 12 cycles to POP on ISR exit
- Up to 120 interrupt sources
  - Includes 16 ARM core specific exceptions
  - The remaining are modules specific
- Up to 16 priority levels fully programmable
  - Reset, NMI and Hard Fault have predefined priority
- Change Interrupt Priority dynamically
- Relocable vector table

Module xyz **Training Outline**

1. Module Overview
2. **On-chip interconnects and inter-module dependencies**
3. Software configuration
4. Typical use cases
5. Demo code explanation
6. Frequently asked question list
7. Reference material

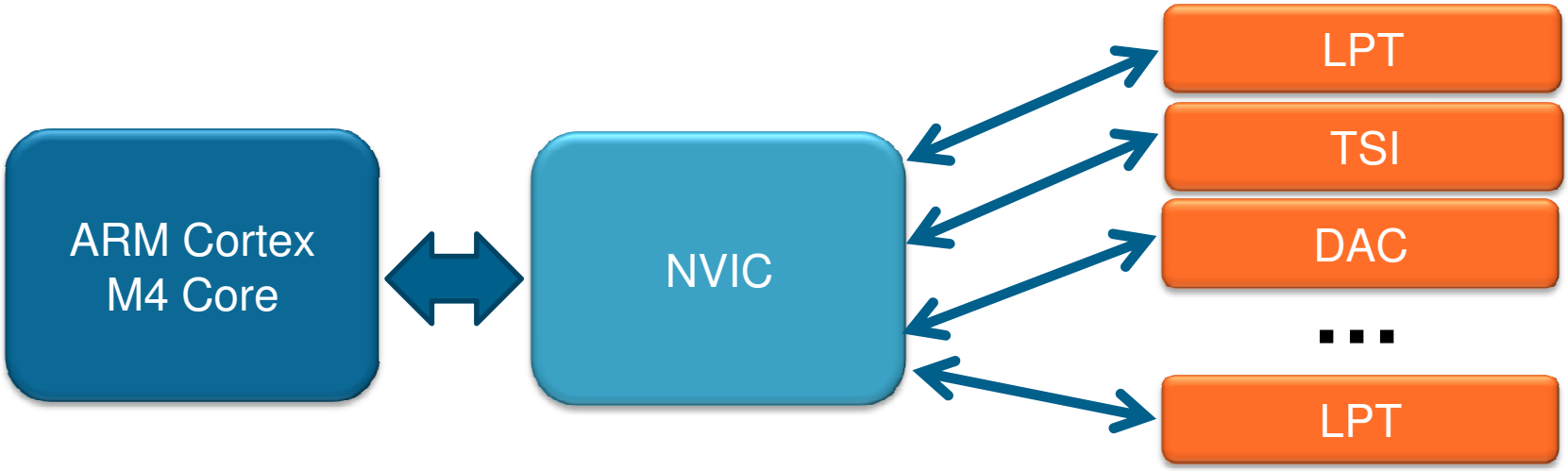
a) Diagram with SoC interconnects
b) Dependency list

freescale™ semiconductor

SoC interconnect diagram

- Any module capable of generating an interrupt will depend on NVIC operation

- NMI can be generated from:
  - External pin (must configure the MUX)
  - Coresight Embedded Trace Buffer (ETB)



freescale™ semiconductor

Module dependencies

►If the MCU is set to Stop or VLPS modes, NVIC will leave the AWIC to manage the interrupts:

- LVD
- ADC
- USB
- ENET
- TSI
- …

►The Cortex-M4 faults will be escalated to jump to the Hard Fault if are disabled:

- Memory Management Fault
- Bus Fault
- Usage Fault

Module xyz **Training Outline**

1. Module Overview
2. On-chip interconnects and inter-module dependencies
3. **Software configuration**
4. Typical use cases
5. Demo code explanation
6. Frequently asked question list
7. Reference material

a) Sequence of register setup
b) Interrupt versus polling configuration
c) Any potential “gotchas” like bit ordering
d) SoC Low power mode considerations

freescale™ semiconductor

Sequence of register setup

- The steps for enabling an interrupt on NVIC:
  1. Enable the peripheral to be used
  2. Set the proper bit on the NVICSERx to enable the interrupt on the NVIC
  3. Clear any pending interrupt by writing to the NVICCPRx to avoid any spurious interrupt
  4. Configure the interrupt priority by writing to the NVICIPxx
  5. Write the ISR
  6. Enable global interrupts

NVIC gotchas

►The NVICIPx just implements the 4 most significant bits , the lower nibble will read as zero and ignore any write.

►Vector number is the next to check if 2 or more interrupts with the same priority are triggered

►Some Low Power modes will shut down the device RAM, you may change the vector table to Flash prior entering to the low power mode to avoid the table to be corrupted.

Module xyz **Training Outline**

1. Module Overview
2. On-chip interconnects and inter-module dependencies
3. Software configuration
4. Hardware configuration/considerations
5. Typical use cases
6. Critical performance specs/Competitor analysis
7. **Demo code explanation**
8. Frequently asked question list
9. Reference material

a) Board level component selection, placement and routing suggestions

Explaining the code

- Set up the LPT interrupt:
  - Locate the interrupt vector that you want on the Vector Table list from the Kinetis device used

Address	Vector	IRQ	Source module	Source description
0x0000_018C	99	83	TSI	Single interrupt vector for all sources
0x0000_0190	100	84	MCG	
0x0000_0194	101	85	Low Power Timer	

  - Find the NVIC Interrupt Set Enable Register (NVICISERx) for your vector:
    - NVICISER2 for the LPT
  - Then take the modulo value of your IRQ number by 32 to calculate which bit to set in the NVICISER2 register
    - 85%32 = 21
    - `NVICISER2 |=(1<<21);//Enable LPT interrupts`

Sequence of register setup

- Clear any pending interrupts from the NVICICPRx
  - Use the same modulo result for knowing which bit to set
  - `NVICICPR2|=(1<<21);  //Clear any pending interrupts on`
- Set the interrupt priority writing to the NVICIPx
  - X is the IRQ number
  - Just the 4 most significant bits are used
  - `NVICIP85 = 0x30;  //Set Priority 3 to the LPT module`
- Write your ISR
  - No compiler directive needed (for IAR tools)

```
void vfnLPT_ISR (void)
{
   LPT0_CSR|=LPT_CSR_TCF_MASK;  //Clear LPT Compare flag
   gu8LPTStatus = 1;
}
```

- Configure LPT to enable interrupts
  - `LPT0_CSR = LPT_CSR_TIE_MASK;  //Enable LPT interrupt`
- Enable global interrupts
  - `EnableInterrupts;`

Module xyz **Training Outline**

1. Module Overview
2. On-chip interconnects and inter-module dependencies
3. Software configuration
4. Hardware configuration/considerations
5. Typical use cases
6. Critical performance specs/Competitor analysis
7. Demo code explanation
8. **Frequently asked question list**
9. Reference material

a) Board level component selection, placement and routing suggestions

NVIC Preliminary FAQ

- **Q: Can I remap the vector table?**
  A: Yes, need to set the base address of the new position on the VTOR register, just set the offset address no other changes needed.

- **Q: What is the latency jumping into my ISR?**
  A: It will take 12 cycles to get into the ISR, if any other ISR is pending, will take 6 cycles to change from one ISR to another.

- **Q: How much priorities the NVIC can manage?**
  A: The NVIC has 16 different interrupt priorities that can be set thru the NVICPRx register

- **Q: Do I need the "interrupt" directive to identify my ISR?**
  A: There is no need of key word on the context of IAR tools. The NVIC will perform the register stacking on HW so no set up previous needed.

Module xyz **Training Outline**

1. Module Overview
2. On-chip interconnects and inter-module dependencies
3. Software configuration
4. Hardware configuration/considerations
5. Typical use cases
6. Critical performance specs/Competitor analysis
7. Demo code explanation
8. Frequently asked question list
9. **Reference material**

a) Board level component selection, placement and routing suggestions

freescale™ semiconductor

NVIC Reference

- ARMv7 Reference Manual
- AN179 “Cortex™-M3 Embedded Software Development” from ARM
- AN209 “Using Cortex-M3 and Cortex-M4 Fault Exceptions” from Keil
- Kxx Reference Manual


freescale™
semiconductor