To set

Example: The **or** operation to set bits 1 and 0 of a register. The other six bits remain constant.

```
#define GPIO_PORTD_DIR_R   (*((volatile unsigned long*) 0x40007400))
GPIO_PORTD_DIR_R |= 0x03; // PD1,PD0 outputs
```

Assembly:

```
LDR  R0,=GPIO_PORTD_DIR_R
LDR  R1,[R0]       ; read previous value
ORR  R1,R1,#0x03   ; set bits 0 and 1
STR  R1,[R0]       ; update
```

c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0	
								value of R1
0	0	0	0	0	0	1	1	
0x03								constant
c_7	c_6	c_5	c_4	c_3	c_2	1	1	
result of the ORR								

To toggle

The **exclusive or** operation can also be used to toggle bits.

```
#define GPIO_PORTD_DATA_R   (*((volatile unsigned long*) 0x400073FC))
GPIO_PORTD_DATA_R ^= 0x80; /* toggle PD7 */
```

Read-Modify-Write Sequence

Assembly:

```
LDR  R0,=GPIO_PORTD_DATA_R
LDR  R1,[R0]       ; read port D
EOR  R1,R1,#0x80   ; toggle bit 7
STR  R1,[R0]       ; update
```

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	
1	0	0	0	0	0	0	0	value of R1
								0x80 constant
~b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	result of the EOR

Delay Function

- **SysCtlDelay()** is a loop timer provided in TivaWare.
- The count parameter is the loop count, not the actual delay in clock cycles. Each loop is 3 CPU cycles.

```
#include "driverlib/sysctl.h"
SysCtlDelay(2000000);  //2000000 loops, each loop
                       //   takes 3 CPU cycles
```

Disassembly:

```
        LDR  r0, [PC, #44]
SysCtlDelay:
        SUBS r0, r0, #0x01     ; take 1 cycle
        BNE  SysCtlDelay       ; take 2 cycles
```


TM4C Clock Sources

- **Precision Internal Oscillator (PIOSC)**
  - Default clock
  - 16 MHz +- 1%
- **Main Oscillator (MOSC)**
  - provides precise, stable clock source
  - 5 MHz – 25 MHz
- **Low-Frequency Internal Oscillator (LFIOSC)**
  - Only used during Deep-Sleep mode
  - 30 KHz
- **Hibernate module Clock Source**
  - 32. 768 KHz oscillator for Hibernate mode

To clear

The **bit clear** operation can be used to clear bits.

```
#define GPIO_PORTE_AMSEL_R (*((volatile unsigned long*) 0x40024528))
GPIO_PORTE_AMSEL_R &= ~0x02; /* clear PE1 */
```

Assembly:

```
LDR  R0,=GPIO_PORTE_AMSEL_R
LDR  R1,[R0]       ; read previous value
BIC  R1,R1,#0x02   ; clear PE1
STR  R1,[R0]       ; update
```

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	
								value of R1
0	0	0	0	0	0	1	0	
0x02								constant
b_7	b_6	b_5	b_4	b_3	b_2	0	b_0	
result of the BIC								

Bit-Specific Addressing

- **If we are interested in bit 2 of Port F (assume initial value on port F is 0x13)**

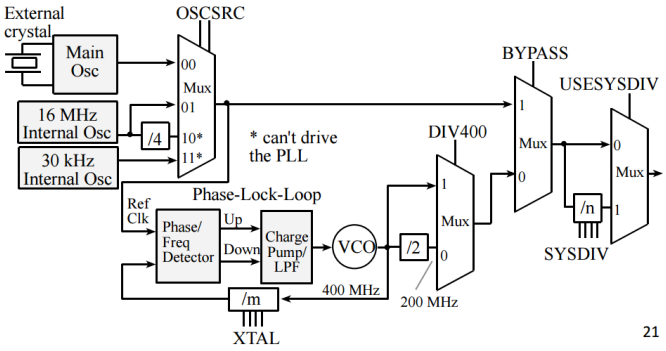
```
#define PF2   (*((volatile unsigned long *)0x40025010))
PF2 = 0x04;  // make PF2 high, other bits will not be modified
Data = PF2; // data = 0x04, only PF2 will be returned, other
            //bits will be 0
```

Address Offset used to specify individual data port bits

If we wish to access bit	*Constant*
7	0x0200
6	0x0100
5	0x0080
4	0x0040
3	0x0020
2	0x0010
1	0x0008
0	0x0004

Clock Control

- Default system clock: 16 MHz ± 1%
- PLL (Phase-Lock-Loop) allows us to speed up or slow down the clock



Clock Control

- Default system clock: 16 MHz ± 1%
- PLL (Phase-Lock-Loop) allows us to speed up or slow down the clock

```
#include "driverlib/sysctl.h"
SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|
SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
```

SYSCTL_SYSDIV_x determines the system clock

System clock = 400 MHz/2/x

Clock Setup

Configure the system clock to ? MHz

SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL |SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);

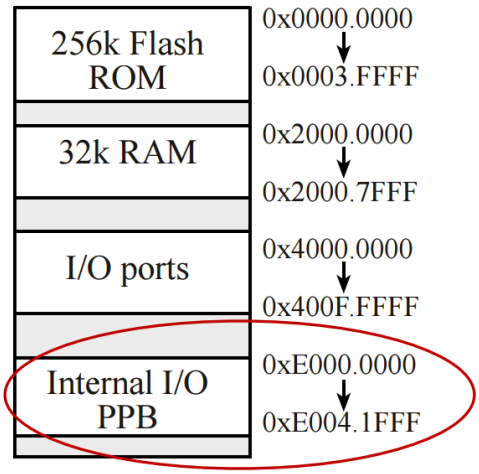
SYSCTL_SYSDIV_5: SYSCTL_SYSDIV_1, SYSCTL_SYSDIV_2, SYSCTL_SYSDIV_3, ...

SYSCTL_USE_PLL: SYSCTL_USE_PLL, SYSCTL_USE_OSC.

SYSCTL_XTAL_16MHZ: SYSCTL_XTAL_4MHZ, SYSCTL_XTAL_4_09MHZ, SYSCTL_XTAL_4_91MHZ,...

SYSCTL_OSC_MAIN: SYSCTL_OSC_MAIN, SYSCTL_OSC_INT, SYSCTL_OSC_INT30, ...

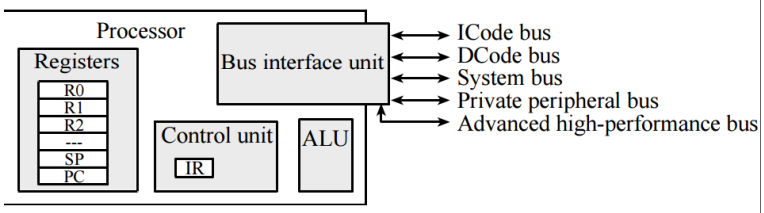
TM4C 123 Memory Address Space

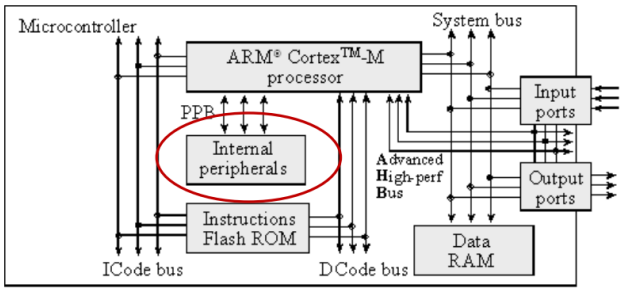


d

dd

ARM Cortex M4 Harvard Architecture





25

Cortex-M4 Interrupts (Table 2.9 Data Sheet)

Vector Number	Interrupt Number (Bit in Interrupt Registers)	Vector Address or Offset	Description
0-15	-	0x0000.0000 - 0x0000.003C	Processor exceptions
16	0	0x0000.0040	GPIO Port A
17	1	0x0000.0044	GPIO Port B
18	2	0x0000.0048	GPIO Port C
19	3	0x0000.004C	GPIO Port D
20	4	0x0000.0050	GPIO Port E
21	5	0x0000.0054	UART0
22	6	0x0000.0058	UART1
23	7	0x0000.005C	SSI0
24	8	0x0000.0060	I²C0
25	9	0x0000.0064	PWM0 Fault
26	10	0x0000.0068	PWM0 Generator 0
27	11	0x0000.006C	PWM0 Generator 1
28	12	0x0000.0070	PWM0 Generator 2
29	13	0x0000.0074	QEI0
30	14	0x0000.0078	ADC0 Sequence 0

27

Cortex-M4 Interrupts (Table 2.9 Data Sheet)

- **Vector Address**: points to the memory location that stores the beginning address of ISR
- **Vector Number (ISR_NUMBER)**: Will be stored in **IPSR** during context switch
- **Interrupt Number (IRQ Number)**: Bit in interrupt registers

NVIC Interrupt Enable Registers

- Five enable registers **NVIC_EN0_R** through **NVIC_EN4_R.**
- The 32 bits in **NVIC_EN0_R** control the IRQ numbers 0 to 31
- The 32 bits in **NVIC_EN1_R** control the IRQ numbers 32 to 63
- ...

Address	31	30	29-7	6	5	4	3	2	1	0	Name
0xE000E100	G	F	...	UART1	UART0	E	D	C	B	A	NVIC_EN0_R
0xE000E104			...						UART2	H	NVIC_EN1_R

Interrupt Conditions

- **Four conditions must be true simultaneously for an interrupt to occur:**

1.NVIC enable (device specific)

2.Arm: control bit for each possible source is set

3.Enable: interrupts globally enabled (I=0 in PRIMASK)

4.Trigger: hardware action sets source-specific flag

- **Interrupt remains pending if trigger is set but any other condition is not true**
  - Interrupt serviced once all conditions become true

Priority Registers on the NVIC

NVIC_PRIx_R

- Each priority register contains an 8-bit priority field for four devices.
- Only the top three bits of the 8-bit field are used. 0-7

Address	31 – 29	23 – 21	15 – 13	7 – 5	Name
0xE000E400	GPIO Port D	GPIO Port C	GPIO Port B	GPIO Port A	NVIC_PRI0_R
0xE000E404	SSI0, Rx Tx	UART1, Rx Tx	UART0, Rx Tx	GPIO Port E	NVIC_PRI1_R
0xE000E408	PWM Gen 1	PWM Gen 0	PWM Fault	I2C0	NVIC_PRI2_R
0xE000E40C	ADC Seq 1	ADC Seq 0	Quad Encoder	PWM Gen 2	NVIC_PRI3_R
0xE000E410	Timer 0A	Watchdog	ADC Seq 3	ADC Seq 2	NVIC_PRI4_R
0xE000E414	Timer 2A	Timer 1B	Timer 1A	Timer 0B	NVIC_PRI5_R
0xE000E418	Comp 2	Comp 1	Comp 0	Timer 2B	NVIC_PRI6_R
0xE000E41C	GPIO Port G	GPIO Port F	Flash Control	System Control	NVIC_PRI7_R
0xE000E420	Timer 3A	SSI1, Rx Tx	UART2, Rx Tx	GPIO Port H	NVIC_PRI8_R
0xE000E424	CAN0	Quad Encoder 1	I2C1	Timer 3B	NVIC_PRI9_R
0xE000E428	Hibernate	Ethernet	CAN2	CAN1	NVIC_PRI10_R
0xE000E42C	uDMA Error	uDMA Soft Tfr	PWM Gen 3	USB0	NVIC_PRI11_R
0xE000ED20	SysTick	PendSV	--	Debug	NVIC_SYS_PRI3_R