MECE336– Microprocessors I

Sleep, Watchdog Timer, EEPROM

Associate Prof. Dr. Klaus Schmidt

Department of Mechatronics Engineering – Çankaya University

Compulsory Course in Mechatronics Engineering
Credits (3/2/4)

Course Webpage: http://MECE336.cankaya.edu.tr

Sleep: Explanation

Sleep Mode

- Use instruction `SLEEP`
- Microcontroller goes into suspended animation
- Power consumption becomes very low (1 μA current)

Entering Sleep Mode

- Watchdog timer is cleared
- Program execution is suspended (does not continue)
- Port settings remain the same
- `PD` bit in `STATUS` is cleared
- `TO` bit in `STATUS` is set
- Timer 0 stops operation

Sleep: Explanation

Leaving Sleep Mode

- External reset through `MCLR` pin
  → Note: it is possible to detect that the microcontroller was in sleep mode because `PD` = 0
- Watchdog timer overflow
  → Microcontroller continues program from the instruction after `SLEEP`
- Occurrence of any enabled interrupt (external, PORTB change, EEPROM write)
  → `GIE` need not be enabled
  → Timer 0 interrupt cannot happen

Sleep: Example

Task

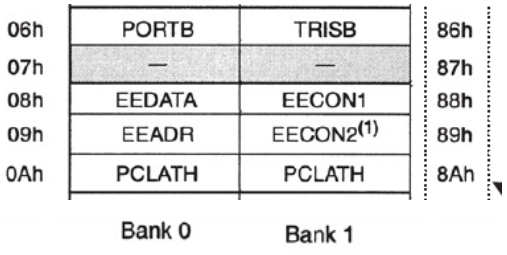
Microcontroller should sleep and only wake up if an interrupt at `RB0/INT` (falling edge) occurs. Then, PORTA should be incremented.

Sleep: Example

EEPROM: Basics

Description

- EEPROM data memory is non-volatile (not erased when power is off)
- EEPROM data memory is readable/writable during normal operation
- EEPROM data memory size is 64 byte
- The following registers are used
  - `EECON1` and `EECON2` (not physical but virtual register)
    $\rightarrow$ `EECON1` and `EECON2` are used to control read/write
  - `EEDATA`: holds 8 bit data for write/from read
  - `EEADR`: holds the address of the EEPROM location to be accessed



	Bank 0	Bank 1	
06h	PORTB	TRISB	86h
07h	—	—	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2(1)	89h
0Ah	PCLATH	PCLATH	8Ah

EEPROM: EECON1 Register

U-0	U-0	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
—	—	—	EEIF	WRERR	WREN	WR	RD
bit 7							bit 0

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit
1 = The write operation completed (must be cleared in software)
0 = The write operation is not complete or has not been started

bit 3 **WRERR:** EEPROM Error Flag bit
1 = A write operation is prematurely terminated
(any $\overline{\text{MCLR}}$ Reset or any WDT Reset during normal operation)
0 = The write operation completed

bit 2 **WREN:** EEPROM Write Enable bit
1 = Allows write cycles
0 = Inhibits write to the EEPROM

EEPROM: EECON1 Register

U-0	U-0	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
—	—	—	EEIF	WRERR	WREN	WR	RD
bit 7							bit 0

bit 1 **WR:** Write Control bit
1 = Initiates a write cycle. The bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.
0 = Write cycle to the EEPROM is complete

bit 0 **RD:** Read Control bit
1 = Initiates an EEPROM read RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
0 = Does not initiate an EEPROM read

EEPROM: Read Operation

Program Structure

Explanation

EEPROM: Write Operation

Basic Write Sequence

Explanation

```
movlw  address;
movwf  EEADR;
movlw  value;
movwf  EEDATA;

bsf    STATUS,5;
bcf    INTCON,GIE;
bsf    EECON1,WREN;
movlw  0x55;
movwf  EECON2;
movlw  0xAA;
movwf  EECON2;
bsf    EECON1,WR;
bsf    INTCON,GIE;
```

EEPROM: Write Example

Task

Write 20 to address 0x04 of the EEPROM. Then, read the content of address 0x14 of the EEPROM.

Sleep EEPROM Watchdog Timer

EEPROM: Write Example

Klaus Schmidt Department
MECE336 – Microprocessors I

EEPROM: Write Example

Watchdog Timer: Basics

WDT Properties

- Free running on-chip RC oscillator (separate from RC oscillator of the `OSC1/CLKIN` pin)
- WDT runs even if the clock of the microcontroller is stopped
- WDT keeps running when executing the `SLEEP` instruction
- WDT time-out (overflow) generates a device reset during normal operation
- WDT time-out makes the program continue after a `SLEEP` instruction
- $\overline{\texttt{TO}}$ bit in `STATUS` is cleared upon WDT time-out
- WDT time-out period is 18 ms without prescaler
- Prescaler with up to 1:128 $\rightarrow$ delay of 2.3 s is possible

Watchdog Timer: Usage

Enable WDT

- Use the relevant configuration bit
  `__config _CP_OFF&_WDT_ON&_XT_OSC`
- Clear the WDT
  `clrwdt` instruction

Relevant Bits in `STATUS`

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 4 **$\overline{TO}$**: Time-out bit
1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
0 = A WDT time-out occurred

bit 3 **$\overline{PD}$**: Power-down bit
1 = After power-up or by the `CLRWDT` instruction
0 = By execution of the `SLEEP` instruction

Watchdog Timer: Example

Task

If a button at `RA0` is pressed in less than 2 sec, `RB0` is turned on for 1 sec and then turned off. Otherwise, a warning LED at `RB1` is turned on for 1 sec.

Watchdog Timer: Example

Watchdog Timer: Example 2

Task

The processor goes into `SLEEP`. It wakes up every 144 msec and checks `PORTA`. If any of the buttons at `PORTA` is pressed, an alarm connected to `RB0` is turned on for 1 sec. Then, the processor goes to `SLEEP` again.

Watchdog Timer: Example 2

Sleep EEPROM Watchdog Timer

Watchdog Timer: Example 2

Klaus Schmidt Department
MECE336 – Microprocessors I