INTEGRATED TECHNICAL EDUCATION CLUSTER
AT ALAMEERIA

SPRING 2015

E-626-A

Real-Time Embedded Systems (RTES)

Lecture #7

Synchronous Serial Links & CCP Modules

Instructor:

Dr. Ahmad El-Banna

Agenda

- SPI
- I2C
- CCP Module

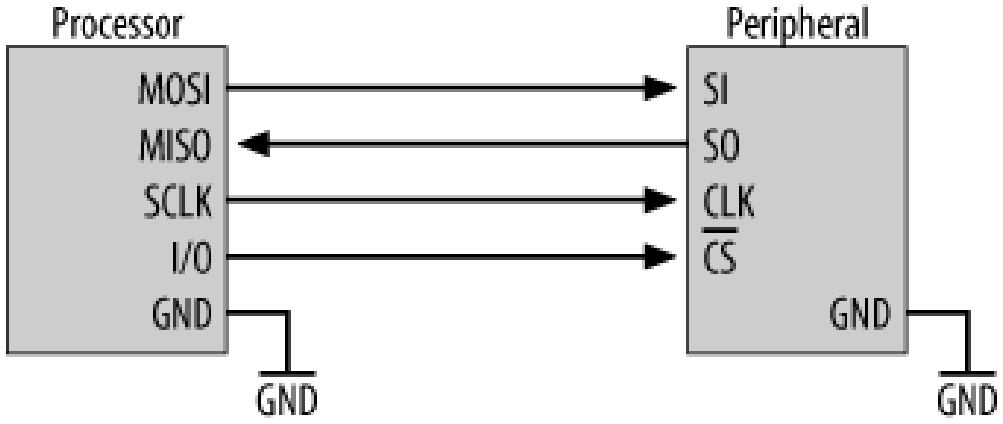
SERIAL PERIPHERAL INTERFACE (SPI)

SPI Interface

- The Serial Peripheral Interface (known as **SPI**) was developed by **Motorola** to provide **a low-cost and simple interface** between microcontrollers and peripheral chips.
- Also known as a **four-wire interface**.
- Used to **interface** to memory (for data storage), ADC, DAC, real-time clock calendars, LCD drivers, sensors, audio chips, and even other processors.
- Unlike a standard serial port, **SPI** is a **synchronous** protocol in which all transmissions are referenced to a common **clock**, generated by the **master** (processor).
- The receiving peripheral (**slave**) uses the clock to **synchronize** its acquisition of the **serial bit stream**.

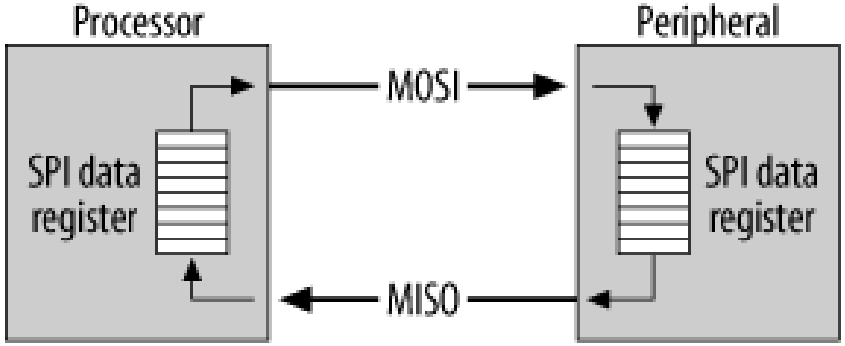
Basic SPI Interface

- SPI uses **four** main **signals**:
  - Master Out Slave In (MOSI)
  - Master In Slave Out (MISO)
  - Serial CLocK (SCLK or SCK)
  - Chip Select (CS)



SPI Transmission

- Both masters and slaves contain a **serial shift register**.
- The master **starts** a **transfer** of a byte **by writing** it **to** its SPI shift **register**.
- As the register transmits the byte to the slave on the MOSI signal line, the **slave transfers** the **contents** of its shift register **back to** the **master** on the MISO signal line.
- In this way, the **contents** of the two shift registers are **exchanged**.
- **Both** a **write** and a **read** operation are performed with the slave **simultaneously**.

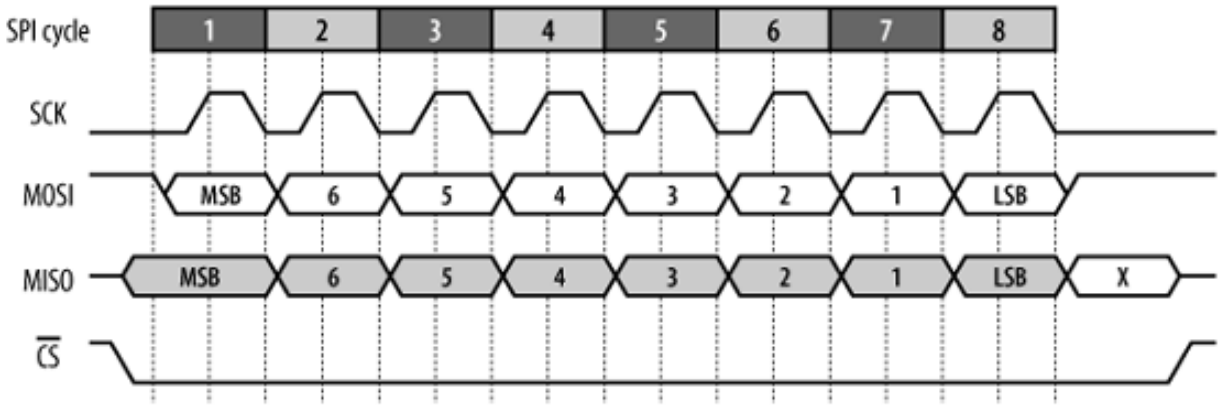


SPI Timing

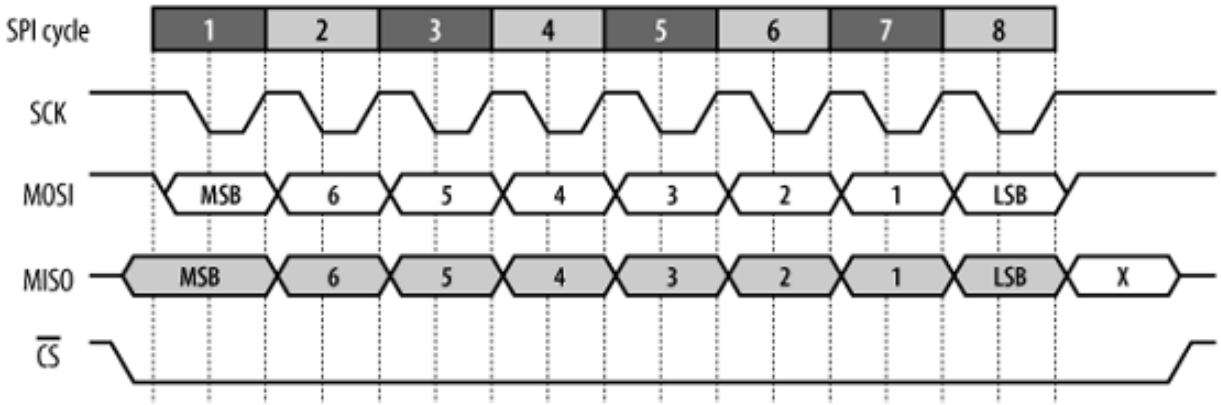
- SPI has **four modes** of operation, **depending** on **clock polarity** and **clock phase**.
- For **low clock polarity**, the clock (SCK) is **low** when **idle** and **toggles** high during a transfer and vice versa.
- The two **clock phases** are known as clock **phase zero** and clock **phase one**.
- For clock **phase zero**, MOSI and MISO **outputs** are **valid on** the **rising edge** of the clock (SCK) if the clock polarity is low and vice versa.

SPI Timing..

SPI timing with clock polarity low and clock phase zero

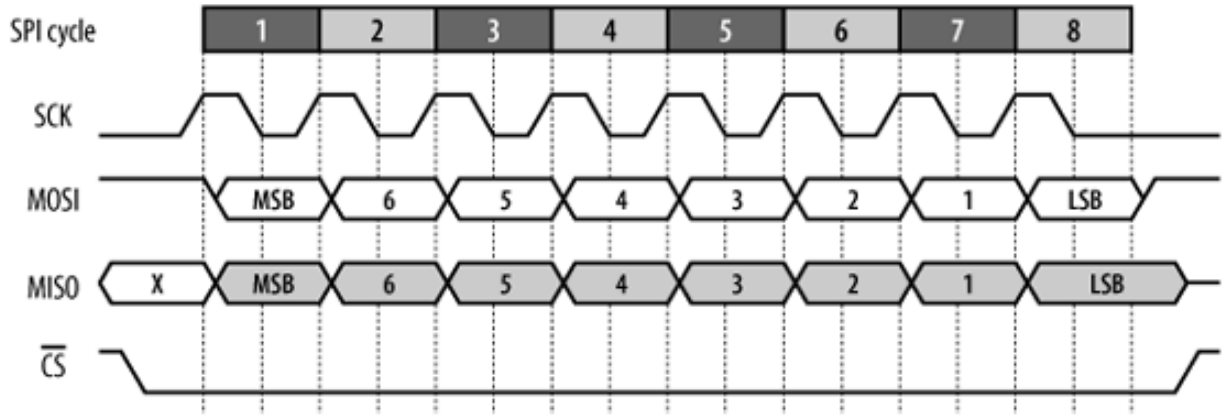


SPI timing with clock polarity high and clock phase zero

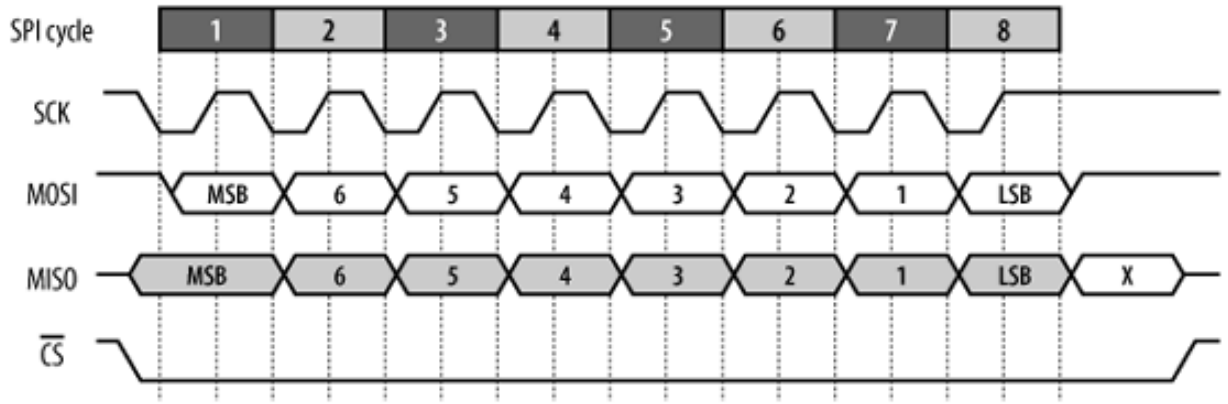


SPI Timing...

SPI timing with clock polarity low and clock phase one

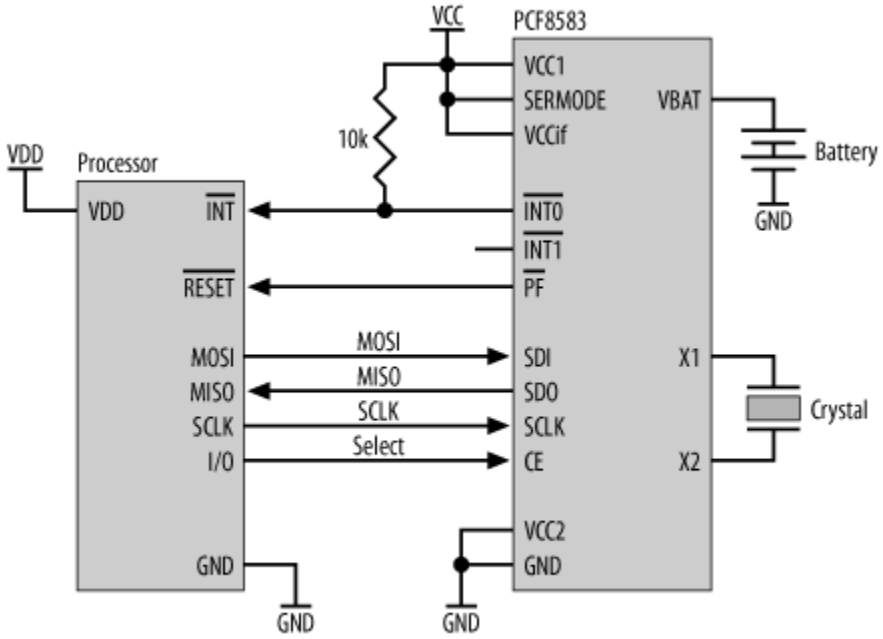


SPI timing with clock polarity high and clock phase one



SPI Application
SPI-Based Clock/Calendar

- A clock/calendar chips contain an oscillator module driven by a crystal, just like a processor.
- The oscillator module ticks over internal counters that track milliseconds, seconds, minutes, hours, days, months, and years.
- They are specifically designed to provide accurate timekeeping, and many have additional functions such as an "alarm" and a watchdog.



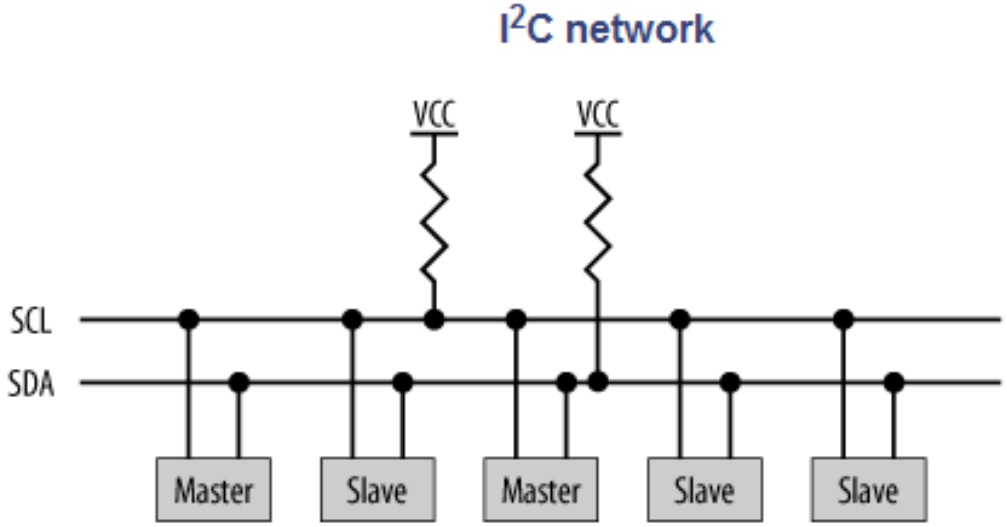
I^2C

Overview of I^2C

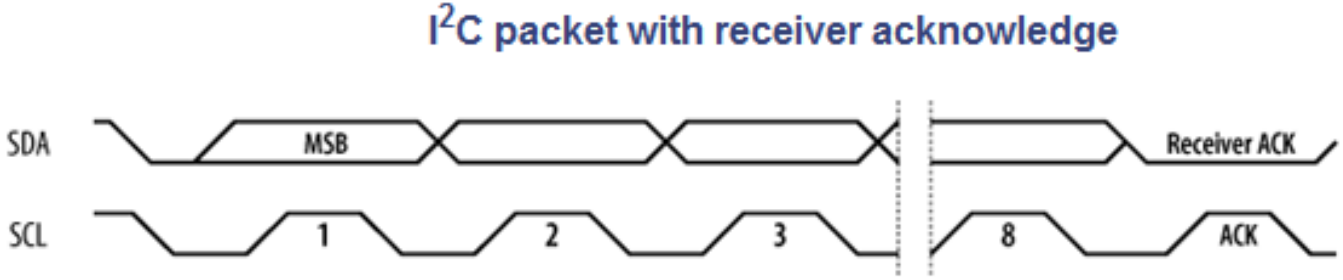
- I^2C (Inter-Integrated Circuit) bus is a **very cheap** yet effective network used to interconnect peripheral devices within **small-scale** embedded **systems**.
- I^2C uses **two wires** to connect multiple devices in a **multi-drop bus**.
- The bus is **bidirectional**, **low-speed**, and **synchronous** to a common clock.
- **Devices** may be **attached** or **detached** from the I^2C bus without affecting other devices.
- The **data rate** of I^2C is somewhat slower than SPI, at **100 kbps** in **standard mode**, and **400 kbps** in **fast mode**.

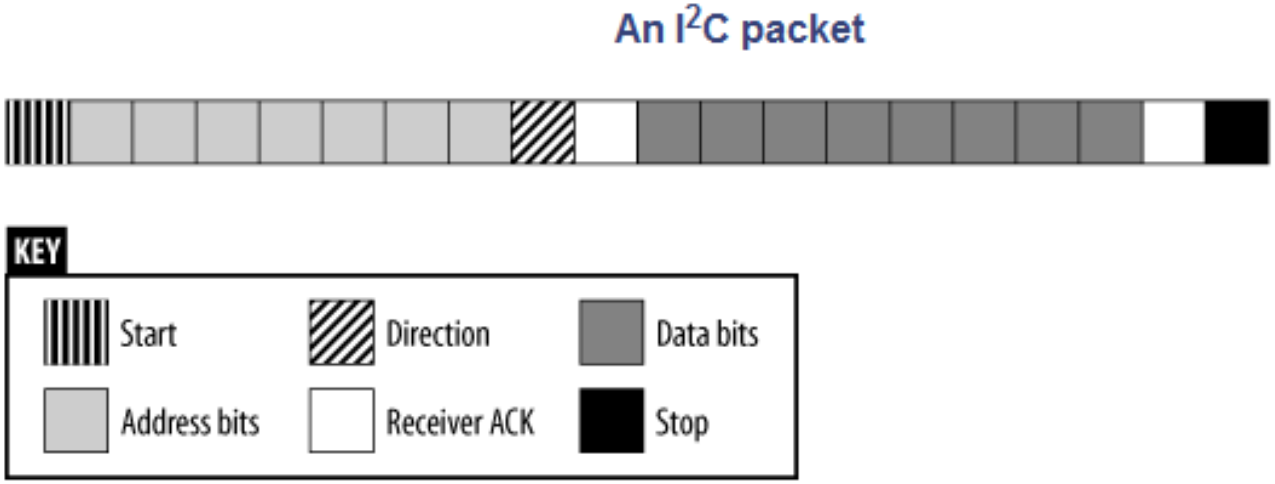
I²C Network

- The **two wires** used to interconnect with I²C are **SDA** (serial data) and **SCL** (serial clock).
- Both **lines** are **open-drain**, they are connected to a positive supply via a **pull-up resistor** and therefore remain high when not in use.
- **Each device** connected to the I²C bus **has** a **unique address** and can **operate** as either a transmitter (a bus **master**), a receiver (a bus **slave**), or **both**.
- I²C is a **multi-master bus**, meaning that more than one device may assume the role of bus master.

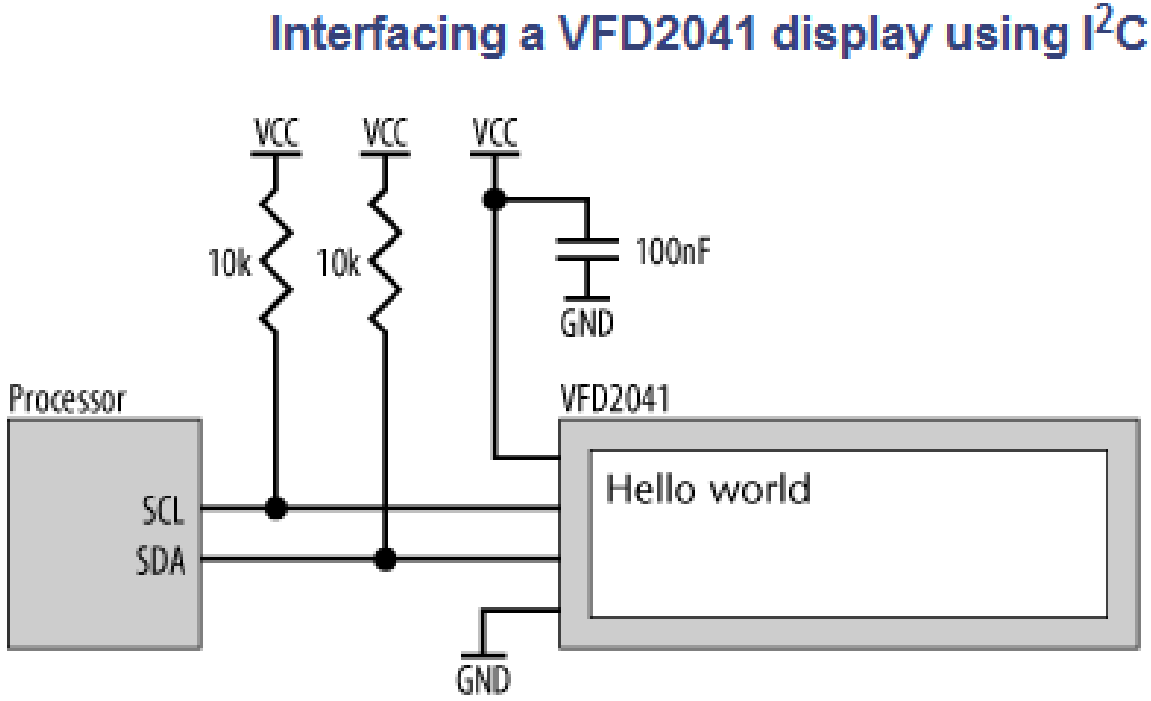


I²C Packets


I²C packet with receiver acknowledge
SDA
SCL
MSB
1
2
3
8
Receiver ACK
ACK


An I²C packet
KEY
Start
Direction
Data bits
Address bits
Receiver ACK
Stop

Adding a Small Display with I²C

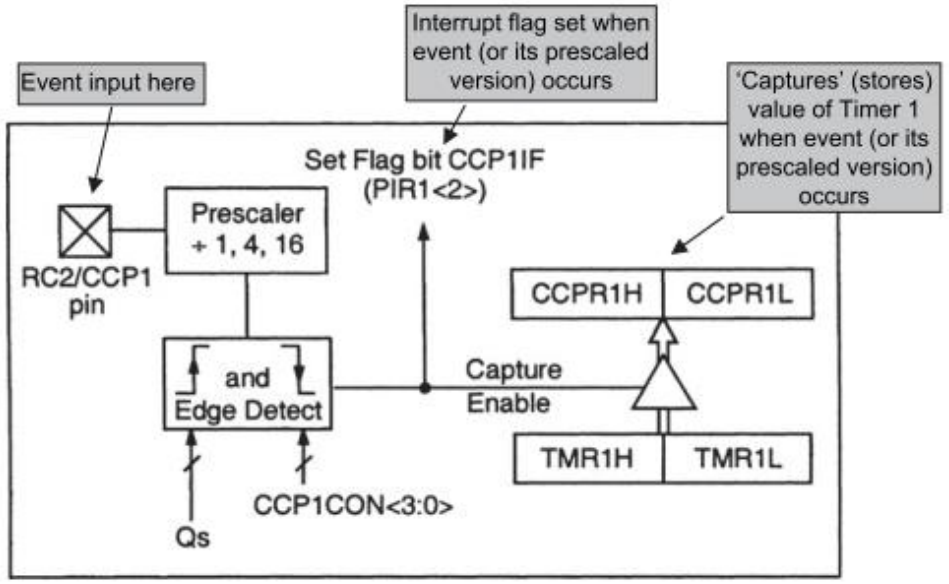


CCP MODULES

Capture Mode

- A **register** that can **record** the **time** of an event is called a '**Capture**' register.
- One that can **generate an alarm** does this by **holding a preset value** and **comparing** it **with** the value of a **running timer**.
- The **16F873A** has **two** such **modules** and they are well worth understanding.
- Each module has two **8-bit registers**, called CCPRxL and CCPRxH, where x is 1 or 2.
- **Together** they form a **16-bit register** that can be used for capture, compare or to form the duty cycle of a PWM stream.
- The CCP modules are **controlled** by their respective CCPxCON **register**

Capture Mode..



U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **CCPxX:CCPxY:** PWM Least Significant bits

Capture mode:
Unused.

Compare mode:
Unused.

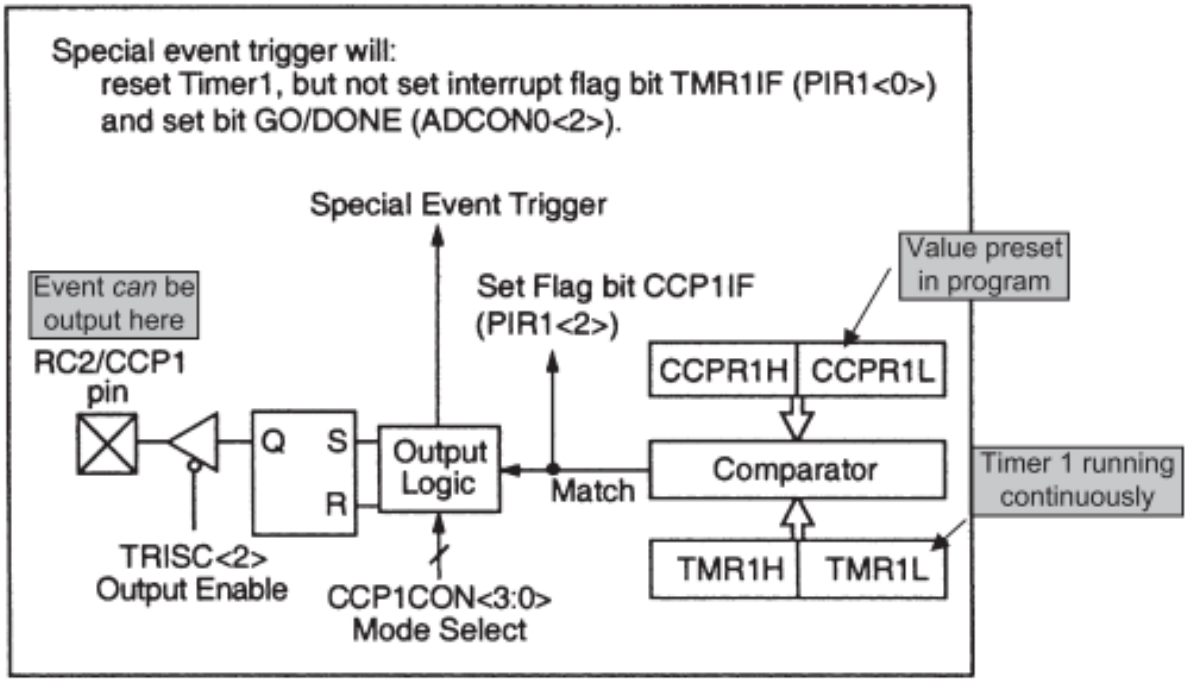
PWM mode:
These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPRxL.

bit 3-0 **CCPxM3:CCPxM0:** CCPx Mode Select bits

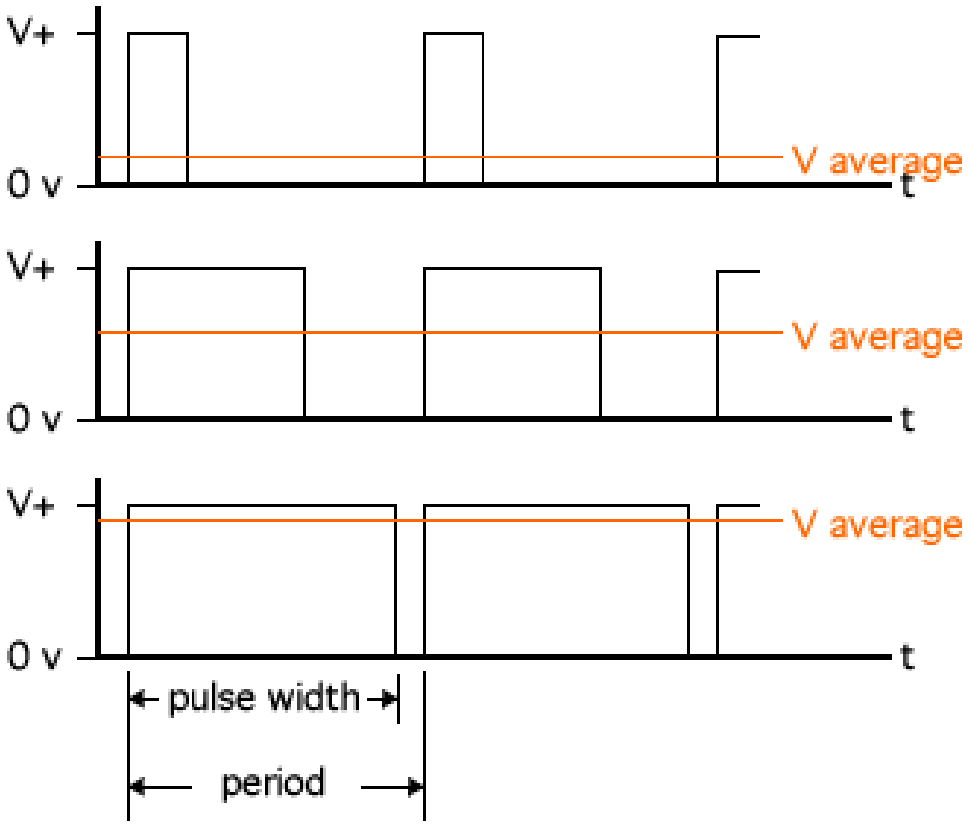
0000 = Capture/Compare/PWM disabled (resets CCPx module)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCPxIF bit is set)
1001 = Compare mode, clear output on match (CCPxIF bit is set)
1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)
1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected); CCP1 resets TMR1; CCP2 resets TMR1 and starts an A/D conversion (if A/D module is enabled)
11xx = PWM mode

Compare Mode

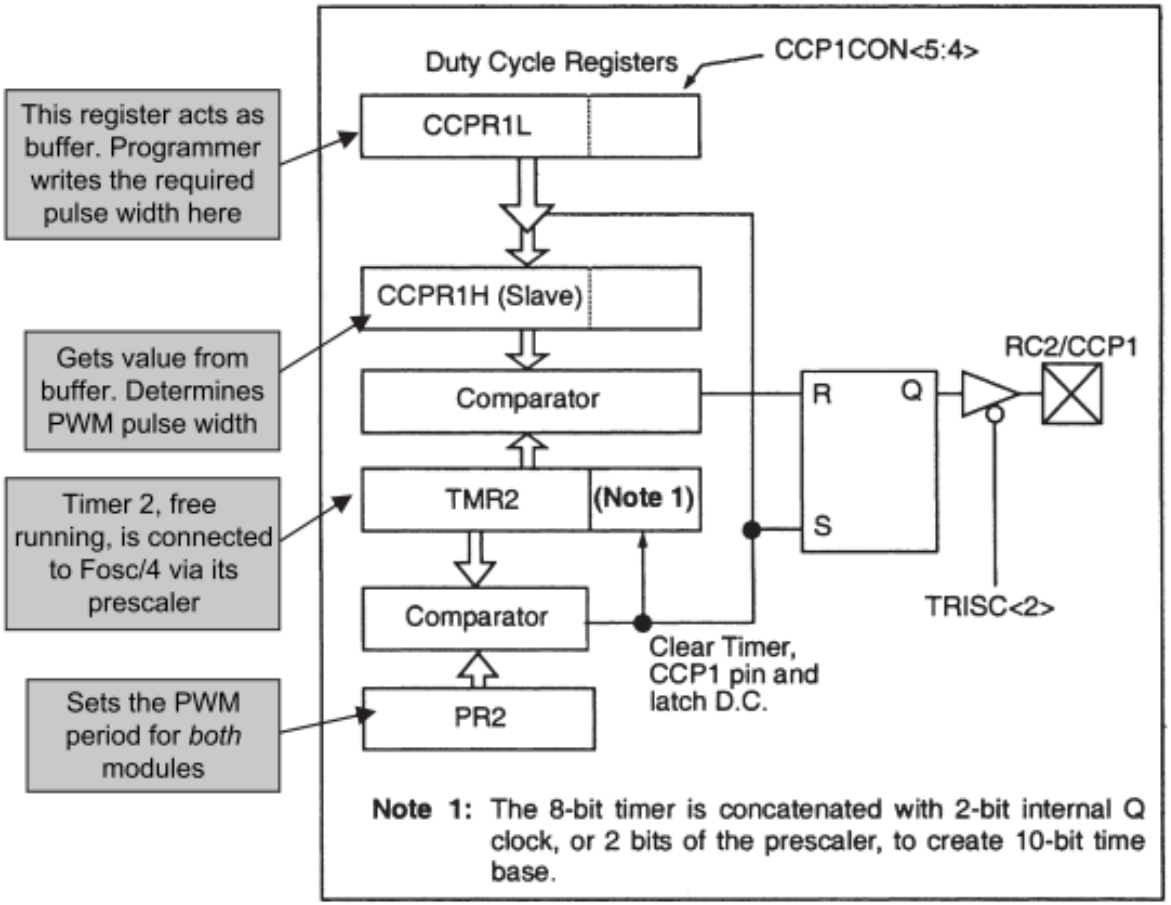
- A digital comparator is designed into the hardware which continuously compares the value of Timer 1 and the 16-bit register made up of CCPR1H and CCPR1L.



PWM, Pulse width modulation



PWM Block Diagram



Sample Proj.

- Apply pulse width modulation with an H-bridge to control a dc motor speed.

- For more details, refer to:
  - Chapter 7,8, **Designing Embedded Hardware**.
  - Chapter 9,10, T. Wilmishurst, **Designing Embedded Systems with PIC Microcontrollers**, 2010.
- The lecture is available online at:
  - http://bu.edu.eg/staff/ahmad.elbanna-courses/12134
- For inquires, send to:
  - ahmad.elbanna@feng.bu.edu.eg