LECTURE 20, 21

Features

Topics to be covered

- Features

Features of 16C6x/7x family.

- Family includes controllers from 16c61/62/64/71/74/710/715 etc.
- They are RISC processors and uses Harvard architecture.
- Different bus widths of data and program memory.Data memory is 8 bit wide where as program memory is 12,14,16 bits wide.The instruction holds immediate data along with instruction code.

Features Contd..

- Only 35 instructions.
- Most instructions take 0.2 microseconds to execute when operated at 20 MHz.
- Machine cycle consist of 4 clock pulses.
- Instruction set is highly orthogonal.
- 1-3 Timers with 8/16 bit prescalar.
- Watch Dog timer (WDT)
- 13-33 I/O pins.

- PIC microcontroller has four optional clock sources.
  - Low power crystal
  - Mid range crystal
  - High range crystal
  - RC oscillator (low cost).
- Programmable timers and on-chip ADC.
- Up to 12 independent interrupt sources.
- Powerful output pin control (25 mA (max.) current sourcing capability per pin.)
- EPROM/OTP/ROM/Flash memory option.
- I/O port expansion capability.
  - Free assembler and simulator support from Microchip at

CPU Architecture

The CPU uses Harvard architecture with separate Program and Variable (data) memory interface. This facilitates instruction fetch and the operation on data/accessing of variables simultaneously.

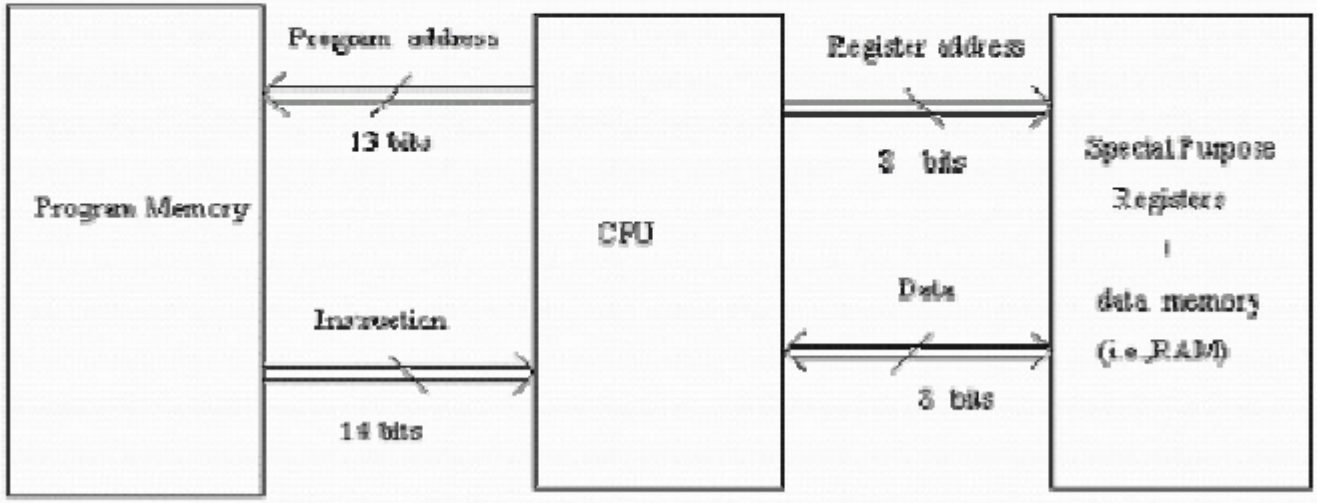


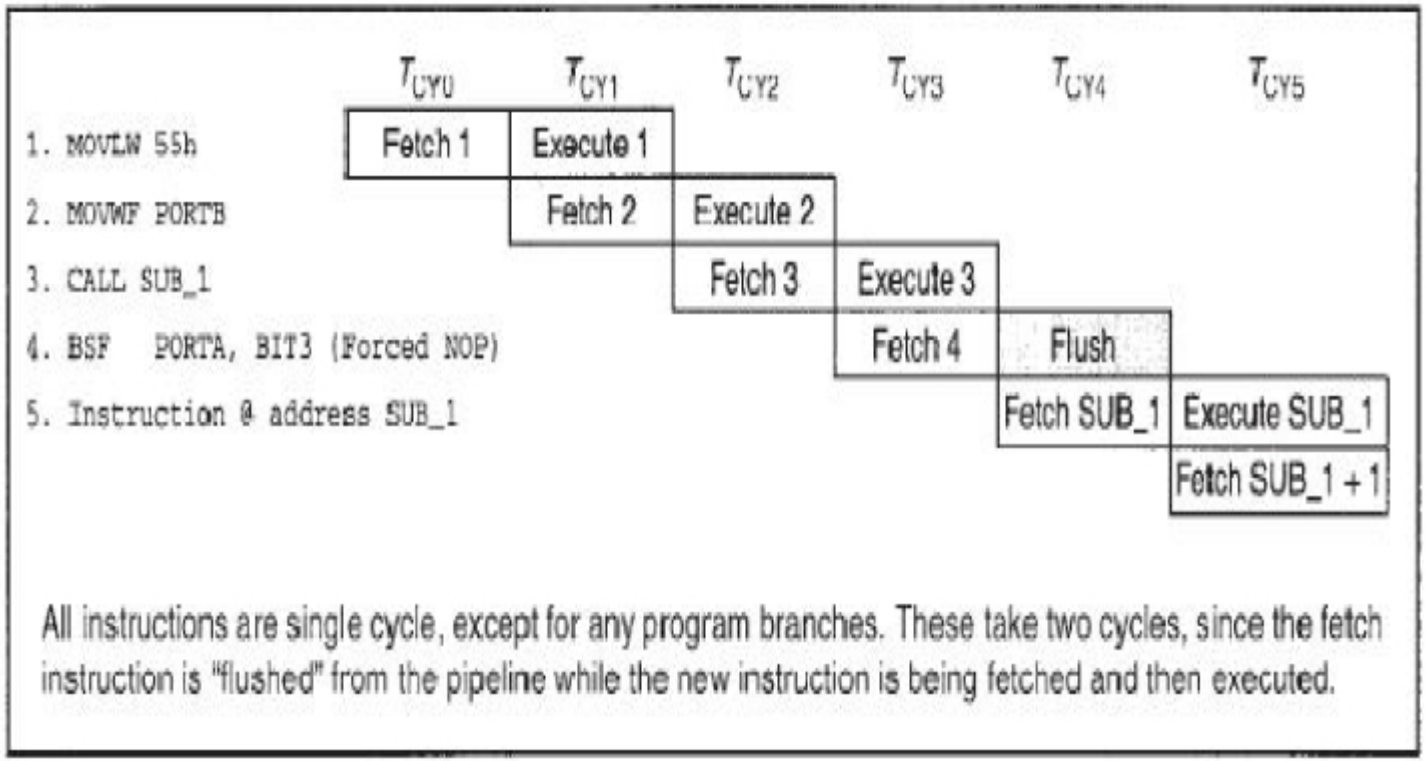
Fig .1 CPU Architecture of PIC microcontroller

Pipelining

The combination of the RISC instruction set and the Harvard memory map used by PIC microcontrollers has an added advantage: instructions can be *pipelined*.
Every instruction in a computer's program memory has first to be fetched and then executed. In many CPUs these two steps are done one after the other—first the CPU fetches and then it executes.
If, however, program memory has its own address and data bus, separate from data memory (i.e., a Harvard structure), then there is no reason why a CPU cannot be designed so that while it is executing one instruction, it is already fetching the next. This is called *pipelining*. Pipelining works best if fetch and execute cycles are always of the same duration, such as a RISC structure gives.
This fairly simple design upgrade gives a **doubling in execution speed!**

All PIC microcontrollers implement pipelining, which is one of the reasons for their comparatively high speed of operation. Each instruction is fetched while the previous one is being executed.

Pipelining fails only for instructions that cause the value in the Program Counter to be changed, for example a program branch or jump. In this case, the instruction fetched is no longer the one needed. The pipelining process must then start again, with the consequent loss of an instruction cycle.



All instructions are single cycle, except for any program branches. These take two cycles, since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.