REGISTER

Nothing more than a bunch of flip flops that store data.

1. General Purpose Registers
2. General Purpose – these have not been explicitly used. Can be used to store ***data***, ***address***.
3. Number of General Purpose registers
4. Number of bits – depends of size of processor (8 bit) register pair(can store 16 bit address)
5. Execution time will be decided by the number of General Purpose registers
6. Size – how many programs to be executed depends on the number of General Purpose registers.
7. Ease of program
8. Special Purpose Registers – used for a specific purpose (User Accessible)
9. Accumulator – used to store one of the operands
10. Program Counter – keeps track of the memory location
11. Status Register – will reflect the outcome of an instruction execution (data overflow, results positive or negative, etc)
12. Stack Pointer – used to implement the data structure (Stack – subroutine calls)

Another way to classify registers is, User Accessible or not.

(Not User Accessible)

1. Memory Address Register (**MAR**)
2. Memory Data Register (**MDR**)
3. Instruction Register (**IR**)
4. Temporary Register (**TR**)

Table 1: REGISTERS

|  |  |
| --- | --- |
|  | ACC (8 bit) |
| B (8 bit) | C (8 bit) |
| D (8 bit) | E (8 bit) |
| H (8 bit) | L (8 bit) |
| Program Counter (PC) (16 bit) | |
| Stack Pointer (SP) (16 bit) | |
|  | S - Z – AC – P – CY  CY 🡪 Carry  P 🡪 Parity  AC 🡪 Auxiliary Carry  Z 🡪 Zed or Zero  S 🡪 Sign (+/-) |

1. Arithmetic Operations (8 bit data)
   1. Addition
   2. Subtraction
   3. Increment
   4. Decrement
2. Logical Operations
   1. AND
   2. OR
   3. XOR
   4. NOT
   5. CLEAR
   6. COMPARE
   7. SHIFT
   8. ROTATE

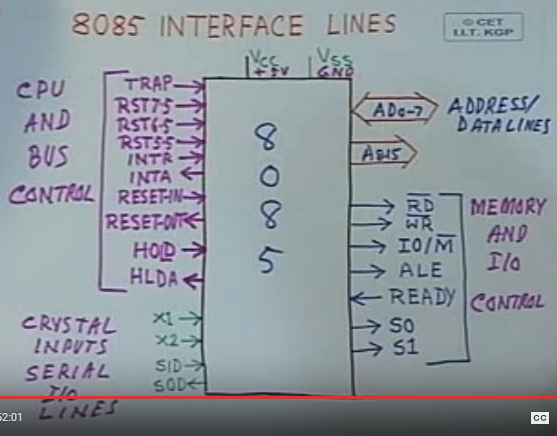
INTERFACE

Nothing more than a bunch of pins that connect to address busses.

1. Memory and I/O Controllers
   1. R/W or rw (linux)
   2. IO/M when line is 1, then I/O process. If, 0 then, Memory process.
   3. Ready/Wait
   4. Address Latch Enable ALE
   5. Status Lines
   6. Address Lines
   7. Data Lines

A through E are necessary. F and G are optional?

1. CPU and Bus Control Lines
   1. Reset
   2. Interrupt
   3. Bus Request
   4. Bus Grant Lines
2. Utility Lines
   1. Power Supply Lines
   2. Vcc Ground
   3. Clock
   4. I/O Lines (sometimes)



TIMING AND CONTROL UNIT

1. Coordinates and Controls all internal devices
2. It is essentially a Finite State Machine (FSM)