**CHAPTER 5 Processor Fundamentals**

von Neumann Model

* There is a processor - the central processing unit (CPU)
* The processor has direct access to memory
* The memory contains a 'stored program' (which can be replaced by another at any time) and the data required by the program
* The stored program consists of individual instructions
* The processor executes instructions sequentially

CPU STRUCTURE

PC:Program Counter - Stores the address of where the next instruction is to be read from

MAR:Memory Address Register - Stores the address of a memory location which is about to have a value read from or written to

MDR:Memory Data Register - Stores data that has just been read from memory or is just about to be written to memory

CIR:Current Instruction Register - Stores the current instruction while it is being decoded and executed

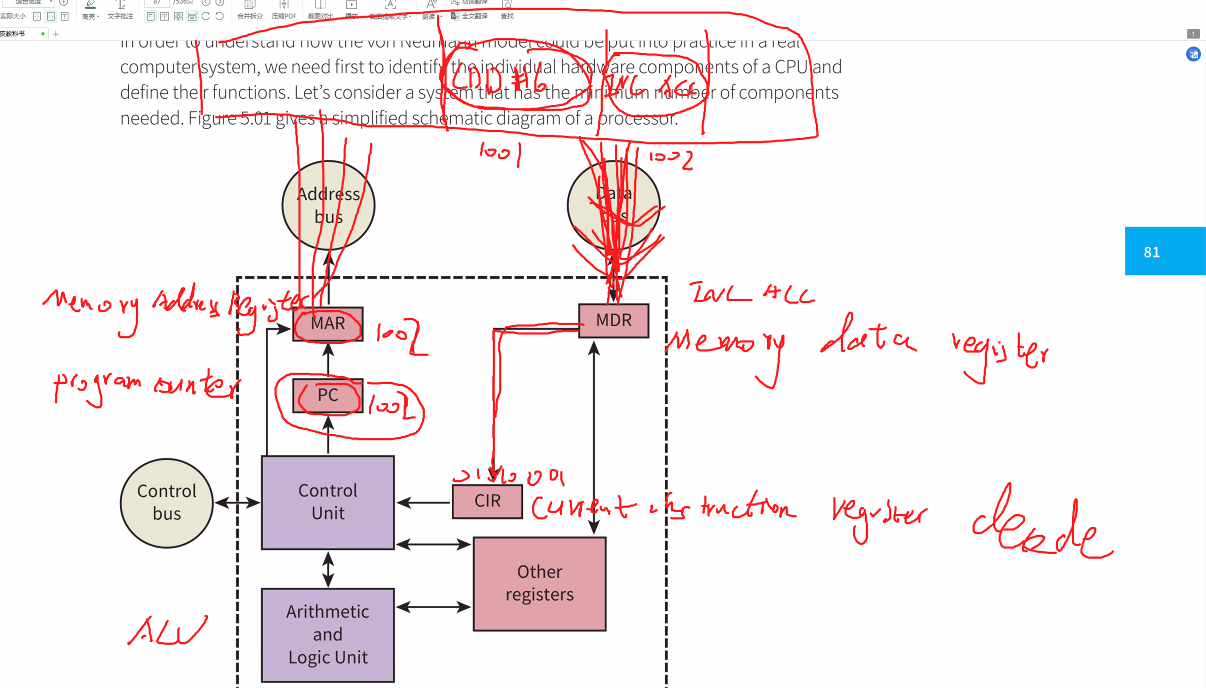
ALU:Arithmetic and Logic Unit

CU:Control Unit

Address bus: a component that carries an address.This can be to the memory controller to identify a location in memory which is to be read from or written to

Data bus: a componet that can carry data from the processor to the memory or to an output device or can carry data from the memory or from an input device, data bus is two-way(bidirectional): it might be carrying data from the CPU to the memory or carrying data to the CPU

Control bus:transfer interrupt signal -> ISR



bidretional:e.g. printer & CPU

monodiretional

Other Reegisters:

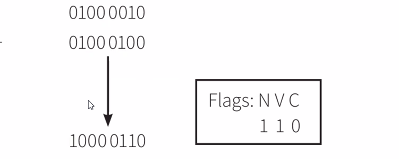
SR:Status Register - each bit of SR used as flag, e.g. to present Negative, Overflow, Curry bit

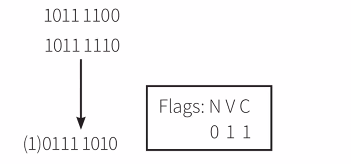
* Flags:

N:Negative

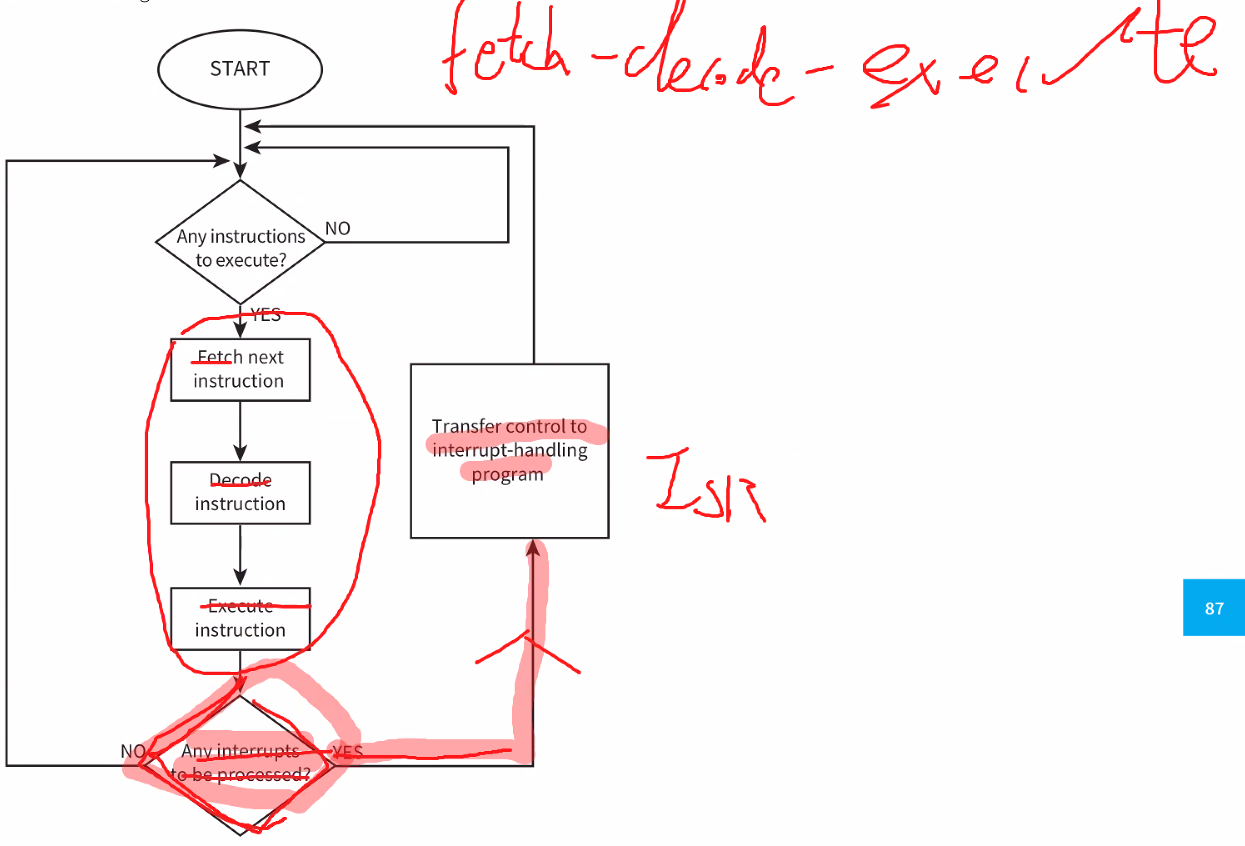
V:Overflow - a condition when the result of a calculation is too large to fit into the number of bits defined for storage

C:Carry bit





The fetch-execute (F-E) cycle



Interrup handling

REASONS

* **a fatal error in a program**
* **a handware fault**
* **a need for I/O processing to begin**
* **user interaction**
* **a timer signal-** Time scheuling:the e.g. of time signal

PROCESSION

* The contents of the program counter and any other registers are stored somewhere safe in memory
* The appropriate interrupt handler or ISR program is initiated by loading its start address into the program counter
* When the ISR program has been executed there is an immediate check to see if further interrupts need handling
* Further interrupts are dealt with by reprated execution of the ISR program
* If there are no further interrupts, the safely stored contents of the registers are restored to the CPU and the originally running program is resumed

System clock:Synchronise operations,creating timing signals

(controls the cycles of activity outside the processor)

EFFECT THE CPU QUALITY

CLOCKS FREQUENCY SPEED - MORE INSTRUCTIONS ARE PERFORMED IN A TIME PERIOD

SIZE OF CACHES - CPU can continue working without waiting for data

NUMBER OF CORES - MORE INSTRUCTIONS ARE PERFORMED IN PARALLEL

Internal clock:controls the cycles of activity within the processor

Register transfer notation

MAR<-[PC]

PC<-[PC]+1;MDR<-[[MAR]]

CIR<-[MDR]

[ ... ]->the content of ...

The content of the MAR is an address; it is the content of that address which is being transferred to the MDR.