2016 MARCH 07 MONDAY

WK 11 DAYS 067 299

Computer Organisation And Architecture

CPU, ALU Data Path, CU, I/O Organisation

⇒ CPU consists of 3 components: ① Registers
② ALU
③ Control Unit

⇒ Some mandatory registers like PC, IR, ACC, MAR etc.

⇒ Only MAR and MBR connected to system bus [MAR to address lines and MBR to data lines], the rest of the registers are connected to internal buses

⇒ Microoperations: Elementary operations taking 1 cycle time to execute for memory reference more than 1 cycle to execute.

⇒ Control signals are required to execute the micro operation.

⇒ Micro program: Sequence of microprograms [every microoperations is]

⇒ Every cycle (phase) of instruction cycle consists of μprogram.

Control Unit

⇒ To generate a sequence of control signals for controlling the microoperations of instructions.

⇒ Requirements of CU design: ① No of control lines (control signals) present
② No of instructions implemented in hardware
③ No of micro operations required for instructions
④ Set of control signals required for each micro operation for each instruction.

Notes

⇒ 2 types of control units:

08 2016 MARCH

TUESDAY DAYS 068-298 WK 11

a) Hardwired CU:

⇒ control signals are implemented in SOP form in terms of instruction and micro operations. Independent hardware

⇒ fastest - used in real time

⇒ not suitable for design and testing as it is not flexible.

⇒ RISC control unit is a hardwired control unit.

b) Microprogrammed CU:

⇒ Control word for every micro operation. Control Data Register is used to hold the control word.

⇒ All control words are stored in control memory.

⇒ Micro-sequencer generates address to fetch next microins[2].

⇒ Address is stored in control address register

⇒ Implemented via ROM i.e. permanent memory.

⇒ Control word normally contains:

a) Branch condition field: $\log_2$(# of branch conditions)

b) Flag field: $\log_2$(# of flags)

c) Control memory address: $\log_2$(Control memory size)

d) Control signals →

(i) Horizontal μprogramming:

⇒ Also known as decoded form of control signal

⇒ Here 1 bit per control signal in control signal field.

⇒ Longer control word and allows parallelism.

(ii) Vertical μprogramming:

⇒ Encoded form of control signals.

⇒ No. of control lines = n suppose, so no of bits = $\log_2 n$

⇒ Use of external decoders required.

⇒ Easy implementation of new instruction but slow.

⇒ Say 2 signals need to be activated same time, so 2 external decoders required.

⇒ Smaller control word as it uses encoded form of CS.

I/O Organisation

⇒ Interrupt Driven, Programmed I/O, DMA

a) Programmed I/O: ⇒ CPU intervention is necessary as it checks the status bits of status register of I/O module.

⇒ When CPU is faster than I/O module, the problem is that CPU has to wait a long time checking the states of I/O module, and this process is known as polling. As a result, it wastes several CPU cycles for this only.

b) Interrupt Driven I/O: ⇒ CPU issues commands to I/O module then proceeds to its normal task until interrupted by I/O device on completion of its work

⇒ Multiple interrupts are handled by:

a) Multiple interrupt lines

b) Software polling: ISR polls I/O devices connected to bus. The first device encountered with IRQ bit set is serviced and the subroutine is invoked.

Notes ⇒ It determines priority but disadvantage is time spent in interrogating the devices

10 2016 MARCH

THURSDAY DAYS 070-295 WK 11

⇒ Vectored interrupt: - The requesting device places its address on the request for CPU and hence polling is overcome
- But priority should be taken care. For that:
a) Daisy chain: The chain of devices connected to same interrupt request line and the chain determines the priority. Demerit is starvation of low priority devices. For that,

b) Bus arbitration: Here daisy chain scheme is extended but with multiple interrupt request - acknowledge lines. Each line group has different priority levels

⇒ Demerit of interrupt-driven I/O: It involves word by word transfer as CPU is involved in transfer, so time taken is larger. This is avoided by DMA, as explained next

⇒ DMA (Direct Memory access):

- Has DMA controller which takes control from CPU.
- 3 tasks process: 1) Initialisation: CPU issues command to DMA with following information which are:
a) Starting address of block b) Address of I/O device
c) Word count: Size of data to be transferred d) Control signal information to read or write e) Control to start DMA transfer.

2) Data transfer then takes place through DMA controller.
3) Finally on completion, DMA controller informs the processor by giving an interrupt signal.

Notes

⇒ For servicing DMA requests, we can do that in DMA breakpoints where CPU needs to use the bus, before that it is suspended for DMA controller operation.

⇒ DMA Configurations : a) Single bus detached DMA : Supports only 1 device, each transfer uses bus twice (I/O to DMA, DMA to memory) and processor suspended twice.

b) Single bus integrated DMA and separate I/O bus. In both cases, multiple device at a time. each transfer uses bus once only (DMA to memory) and processor suspended once.

⇒ Types of DMA Transfer Modes :

a) Burst mode / block mode DMA : ⇒ Size of block predetermined.
- Block as a whole is transferred in 1 go.
- DMA interrupts CPU after entire block transfer.

b) Cycle Stealing Mode : Demerit of above is CPU may be inactive for long time during block transfer phase.
- Here DMA requests repetitively for each byte till transfer completes
- So basically transfer is on word by word basis.
- Advantage is CPU remains inactive for smaller time but yes it is still idle during DMA transfer phase.

c) Demand / Continuous / Complete Transfer DMA : Similar to burst mode DMA only the difference is entire data is transferred instead of a data block.

d) Transparent (Hidden) DMA : - Here DMA does not disturb CPU
- It uses those cycles where CPU does not require bus, as in instruction decode and execute phase.

NOTES: Let X = Preparation time (Depends on I/O speed) , Y = (Depends on main memory speed)

∴ % of time CPU is blocked = $\left(\frac{Y}{X+Y}\right) \times 100$

% of time CPU is busy = $\left(\frac{X}{X+Y}\right) \times 100$