Class Test -01

Shahul Hameed. S
1KN18CS094
CSE 'A' Sec '4th Sem
MES

① Difference b/w RISC and CISC

>

RISC	CISC
* Reduced instruction	* Complex instruction
* Executes within single clock cycle	* Executes as in more clock cycle.
* High clock speed	* low clock speed.
* It has large general purpose registers set	* no load and store registers
* Instructions executed in parallel by pipline	* no such pipelines.

② Explain pipelining with example.

> A pipeline is the mechanism a RISC processor uses to executed Instructions.

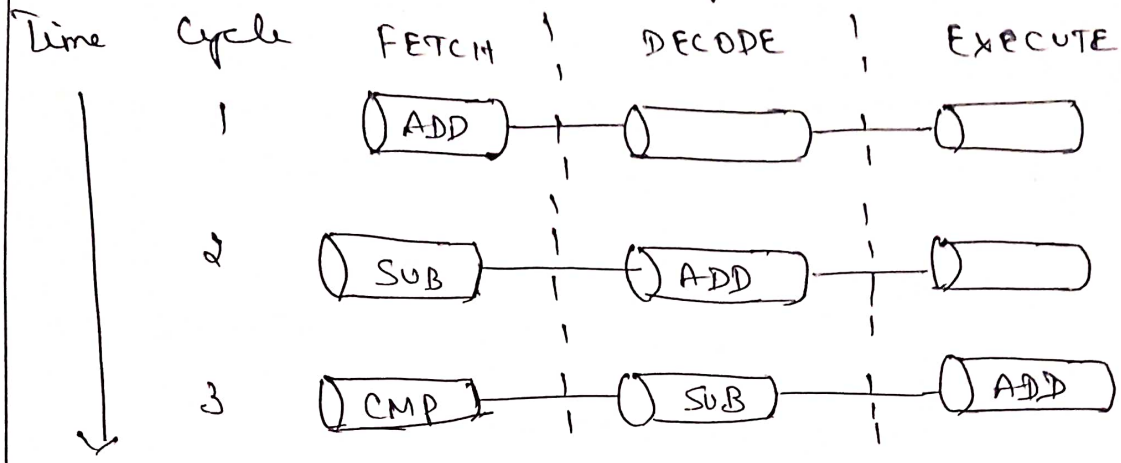
Pipeline speeds up execution. Here next Instruction is fetched while other instructions are being decoded and fetched.

ARM three stage pipeline

* Fetch loads an instruction from memory

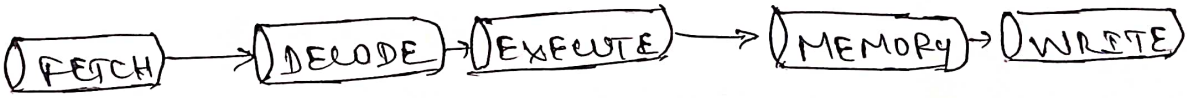
* Decode identifies an instruction to be executed.
* Execute processes the instruction.

⇒ Pipelined Instruction Sequence



Here 3 instructions are placed in pipeline sequence.
In first cycle ADD instruction is fetched.
In second cycle ADD is Decoded SUB instruction is fetched
In third cycle CMP is fetched, SUB is Decoded and ADD is Executed.

ARM9 FIVE stage pipeline



ARM9 Increases 2 stages by adding memory and write back stage.

ARM 10 increases the length of pipeline by adding 6th stage issue.

Class Test - 02

Shahul Hameed. S
1KN18CS097
CSE 'A' sec 2nd sem
MES

(1) Explain the MOV instruction set provided by ARM with the Example for each.

> This instruction copies N to Rd; where N is a register or intermediate value. It is useful for setting initial value and transferring data b/w Registers.

Syntax: <Instruction> {<cond>} {S} Rd, N

MOV - Move a 32 bit value into a register

MOV - Move the NOT of 32 bit value into a register

Rd = ~N

Eg:

```
MOV r7, r5
PRE r5 = 5
    r7 = 8
MOV r7 r5 ; let r7 = r5.
After r5 = 5
      r7 = 5
```

(2) Explain PSR instruction with example.

> * The ARM instruction set provides 2 instructions to directly control a program status registers.
> * The MSR instruction transfer the contents of a registers into the CPSR or SPSR
> * The MSR instruction transfer the contents of either the

cpsr or spsr into register.

* Together these instructions are used to read and write the cpsr and spsr.

Syntax:

MRS {<cond>} Rd, <cpsr/spsr>

MSR {<cond>} <cpsr/spsr> - <fields>, Rm

MSR {<cond>} <cpsr/spsr> - <fields>, # immediate.

Class Test-03

MES

Shahul Hameed. S
1KN18CS097
CSE 'A' Sec 4th Sem

(1) Difference b/w general computing System and Embedded computing System.

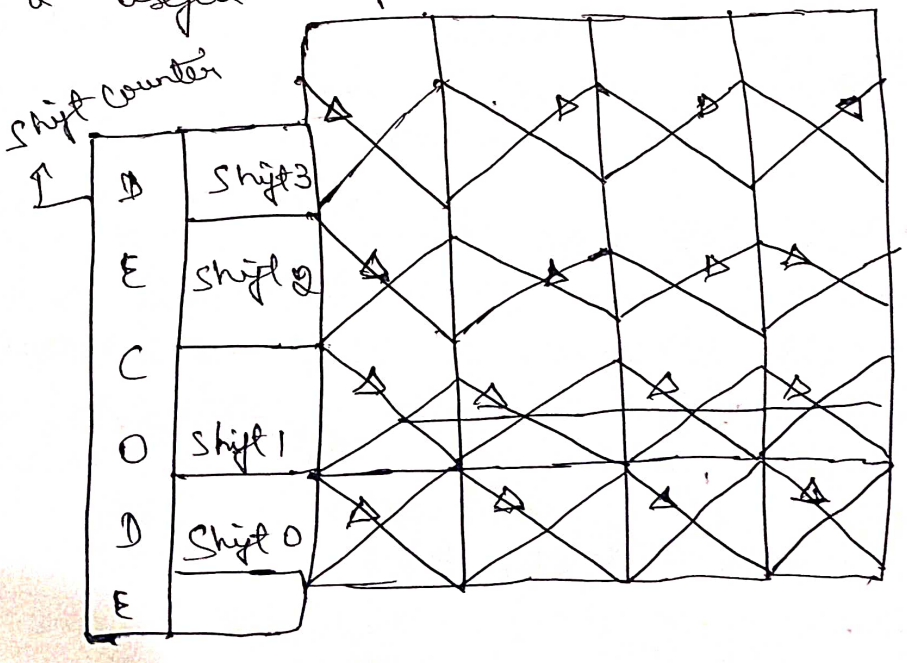
General computing System	Embedded computing System.
* A system which is a combination of a genius hard purpose OS for executing a variety of applications.	* A System which is a combination and embedded OS for executing and efficient of applications.
* Contains a general purpose OS (GPOS).	* May or may not contain an OS for functioning.
* Response requirements are not time critical	* For certain category of Embedded Systems like machines critical Systems, the response time requirement is critically high.
* Need not be deterministic in Execution behaviour	* Execution behaviour is deterministic for certain types of Embedded Systems like μ and Real time Systems.
* Applications are programmable by the users	* The firm ware of the Embedded System is pre programmed and it is non-alterable by the End user.

(9) What is Barrel Shifter? Explain with an example.

A barrel shifter is a digital circuit that can shift a data word by a specified number of bits without the use of an sequential logic, only combinational logic is used. one way to implement a barrel shifter is a sequence of multiplexers, where the o/p of one multiplexers is connected to the i/p of the next multiplexer in a way that depends on the shift distance. A barallel shifter is often used to shift is rotates in bits in modern microprocessor typically within single clock cycle.

For Eg:- Take a 4 bit barrel shifter within i/p A, B, C, D The shifter can cycle the order of the bits ABCD as DABC, CBAB or BCDA in this case, no bits are lost. That is it can shift all of the o/ps upto 3 position to the right.

The barrel shifter has a variety of applications including being a usefull component in a microprocessors.



Class Test - 04

MES

Shahul Hameed. S
1KN18CS097
CSE' A' sec 4th sem

(1) what are the operational and non-operational quality attributes of an embedded system?

> operational quality attributes are present the relevant quality attributes released to the embedded system when it is in the operational mode online mode.

(1) Response: is a measure of quickness of the system. It gives an idea about how fast your system is tracking the changes in i/p variables.

(2) Throughput: deals with the efficiency of a system. Throughput is defined as the rate of production / operation of a defined process over a stated period of time.

(3) Reliability: is a measure of how much percent you can surely upon the proper functioning of the system its failures.

(4) maintainability: deals with support and maintainance to the end user / client in case of technical issues and product failure on the basis of a routine system checkup.

(5) Security: aspect carries, confidentiality integrity and the availability.

Non-operational quality:

(1) testability and Debug ability: deals with how easily one can test his/her design, application and by which means he/she can test it.

(2) Evolvability: is a term which closely related to the case

with which the embedded product can be to take advantages of new firmware technologies.

(3) Portability:- An embedded product is said to be portable if the product is capable of functioning as such in various environments.

(4) time to prototype and market:- is the time elapsed between the Conceptualisation of a product & the time at which the product is ready to selling / use.

(5) per unit and total cost:- is a factor which produce is closely monitored by both end user and product manufacturing.

(2) Explain the fundamental issues in hardware, software, co-design

* Selecting the model:- In hardware software co-design models are used for capturing and describing the system characteristics.

* Selecting the architecture:- The architecture specifies how a system is going to implement in terms of the number and types of different components and the interconnection among them.

* Secreting the languages:- A programming language captures a 'computational model' and maps it into architecture.

* partitioning system reqs into hardware and software:- It may be possible to implement the system requirements in either hardware / software.

Class Test - 05

MES

Shahul Hameed. S
1KN18CS097
CSE A' Sec 4th Sem

(1) Differentiate b/w hard real time and soft real time OS with a eg. for each.

Hard real time	Soft real time
* In hard real time System the size of data file is small / medium.	* In Soft real time system the size of data file is large.
* In this System Safety is critical	* In this system, Safety is not critical.
* In this system response time is in millisecond	* In this system response time are higher.
* A hard real time System is very restrictive.	* A soft real time System is less restrictive.
* Satellite launch	* DVD player.

(2) Explain the simulator and Emulator

Simulator :- A simulator is designed to create an environment that contains all of the software variables and configurations that will exist in an applications actual production environment. Simulators don't attempt to emulate the actual hardware that will host the application in production. Because simulators create only software environments, they can be implemented using high-level program

language.

Emulator :-

This doesn't attempt to mimic all of the hardware features of a production Environment as well as software features. To achieve this, you typically used to write an emulator using assembly language. Emulator may not do a perfect job of Emulating the hardware and software of a production environment, they are not a substitute for real-device testing. They just allow you to set up an environment.