Name - Md. Rakib Hasan.
ID - 2216080021
Batch - 08 (CSE-EV)
Subject - Computer Architecture.
Code - CSE 313.

Answer to the Question-No - 01 - (a)

⇒ Ans: (a) Describe possible Organization of Embeded System:

As its name suggests, Embeded means Something that is Attached to Another thing. An Embeded System Can be though Of as Computer Hardware System having Software Embeded init. An Embeded System Can be an ~~independent~~ Independet System or it can be a part Of a large Systame. An Embeded System is a Micro controller Of Microprosseson based System Which is designed to perform a Specific task. for Example, a fire Alarm is an Embeded System. it will Sense Only Smoke.

⇒⇒ An Embeded System has three Components —

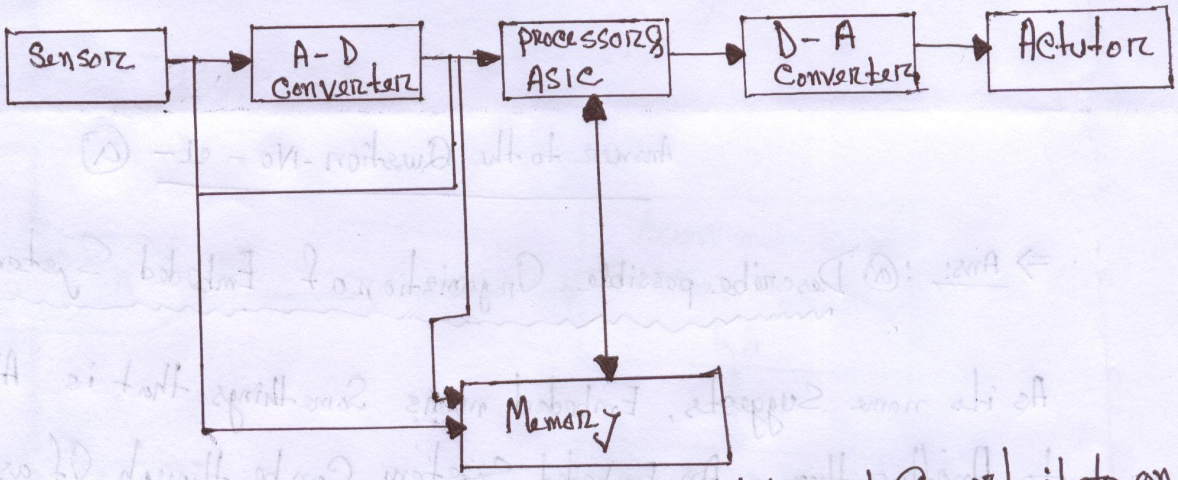
⇒ 1: It has Hardware.
⇒ 2: It has application Software.
⇒ 3: It has Real time Operating System. (RTOS). that Application Software an provide mechanism to let the prosseson run a prosses as per Scheduling by following a plan to Control the latencies.

P.T.O

1(a)

⇒ Basic Structure of an Embeded System :

⊞ The following illustration shows the Basic Structure Of an Embeded System —:



⇒ Sensor: It measures the physical Quantity and Converts it to an Electrical Signal wich Can be read by an Observer or by any Electronic Instrument like an A2D Converter. A Sensor Stores the Measured Quantity to the Memory.

⇒ A-D Converter: An analog-to-Digital Converter Converts the Analog Signal Send by the Sensor into a Digital signal.

⇒ processor& Asic processors prosses the Data to mesure the Output And Stor it to the memory.

⇒ D-A Converter: A Digital-to-Analog Converter Converts the Digital Data fed by the processor to Analog Data.

⇒ Actuator An Actuator Compares the Output given by the D-A Converter to the Actual (Expected) output Stored in it And stores the Approved Output.

Answer to the Question No - 1 - (b)

(b) Answer: Drawing the pass one And two in a two-pass Assembler.

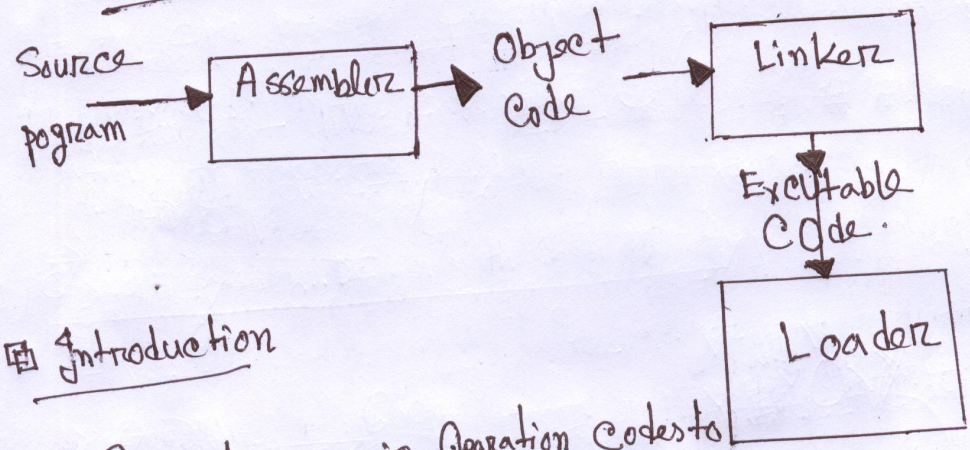
pass I of the Assembler

Data Structure used in pass I

1. OPTA
2. SyMTAB
3. LITTAB
4. POOLTAB.

Algorithm - Intermediate Code - Declaration And Assembler Directive - processing.

⇒ ⇒ pass II of the Assembler:



Introduction

→ Convert mnemonic Operation codes to their machine Language Equivalents.
→ Convert symbolic Operands to their Equivalent machine Address.
→ Build the machine Instruction in the proper format.
→ Convert the Data Constant to internal machine representation
→ write the Object program and the assembly listing.

(b)

⇒ ⇒ Two pass Assembler Diagram—

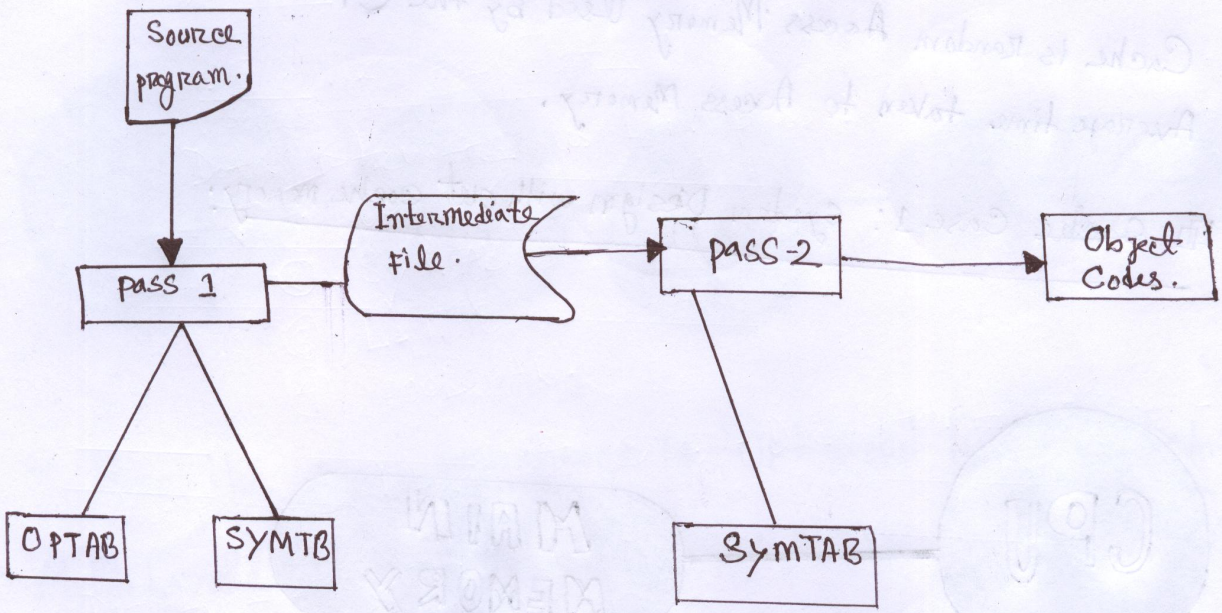


Fig: Two pass Assembler.

⇒ pass 1:

① separate the symbol, Mnemonic Op code, And Operand fields.
② Build the symble table.
③ perform LC processing.
④ Contruct Intermediate Representation.

⇒ pass-2 synthesize the Target program.

Advanced Assembler Directives.

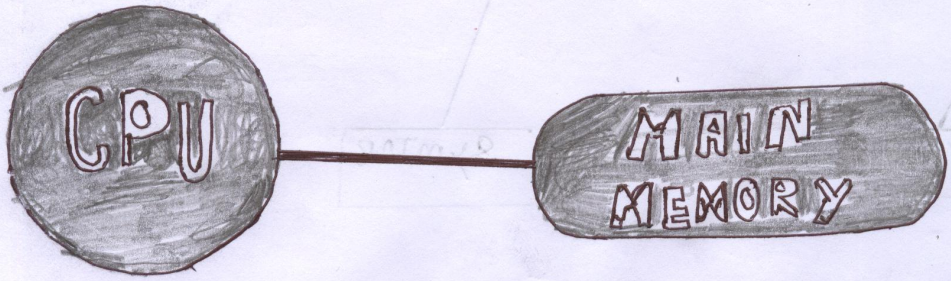
→ ORIGIN.
→ EQU.

Answer to the Question-No-1-C

© Ans: Three Level Cache Organization:

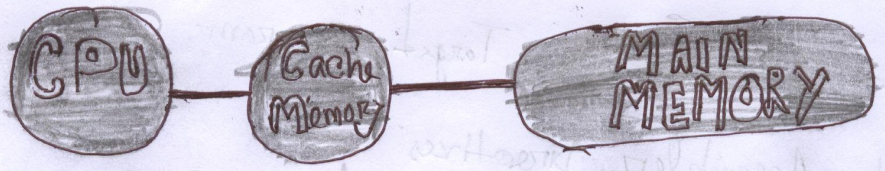
Cache is random Access Memory Used by the cpu to reduce the Average time taken to Access Memory.

Cache Case 1: System Design with out cache memory:



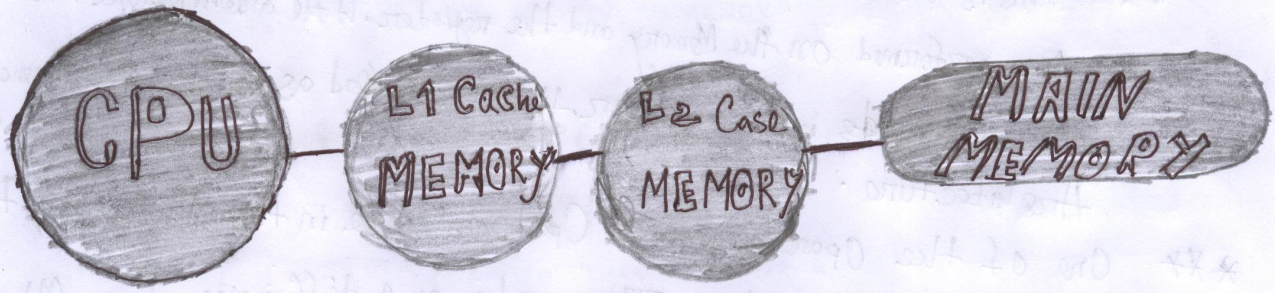
Here the CPU Directly Communicates with the main Memory and No Caches are involved. In this case, the CPU needs to Access the main Memory. 10 times to Access the Desired Information.

Case 2: System Design with Cash Memory:



Here the CPU at first checks whether the desired data is present in the Cache Memory or not i.e. whether there is a "hit" in cache or "Miss" in the cache. Suppose there is 3 miss in Cache Memory. then the main memory well be Accessed Only 3 times. We can See that here the miss penalty is reduced. Because the main Memory is the Accessed a lesser number of times then that in the previous case.

Case-3: System Design With Multi level Cache Memory:



→ Here the case performance is optimized Furthure by introducing Multi Lavale Cases. As swon in the above figure, we are Considering 2 Lavale Case Design. Suppose there is 3 miss in the L1 Cache ~~We are Co~~. & Memory and Out of this 3 missed there is 2 miss in the L2 Cache memory then the main memory will be accessed. Only 2 times. It is clear that Here the miss penalty is Reduced Considerably then that in the previous case thereby improving the performance of ~~cach~~ Cache Memory.

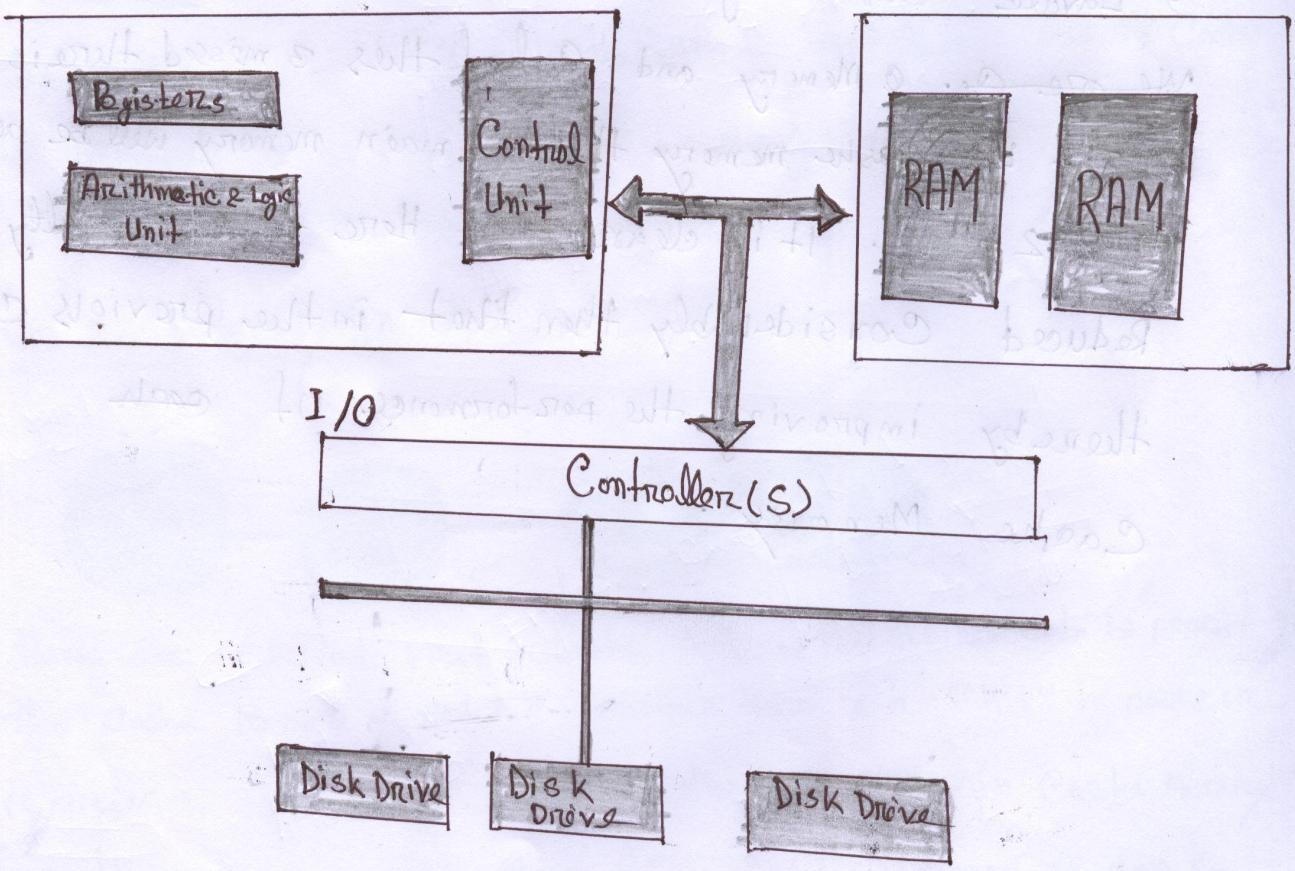
Answer to Question No-2- (a)

=> (a) Memory Access Register:

** This is an architecture that is led by Instruction so that Operations Are performed on the Memory and the register. If the architecture has all the Operands in the register. then it is called as register plus memory Architecture.

*** One of the Operands of an Operation Can be in the Memory and the Other one in the register. This acts as a difference from Other architecture where both Operands of Operation Should be either in the register or in the Memory.

** Example are * IBM system/360 and Intel X86.



(2)(a)

** Registeres are small in size and the number are also less in CPU. The size of a register is less then 64 bits. It is faster then the main Memory and disk Memory. The word size Depends on the size of General-purpuse Registers.

⇒ Type and function Of Register Memory.

- Memory Address Register:
- Memory Buffer Register
- Instruction "
- program Counter register.
- Accumulator Register.
- stack Controler "
- flag Register

Answer to the Question - No - 2 (b)

2 (b) Ans: Draw the Cache read Operation:

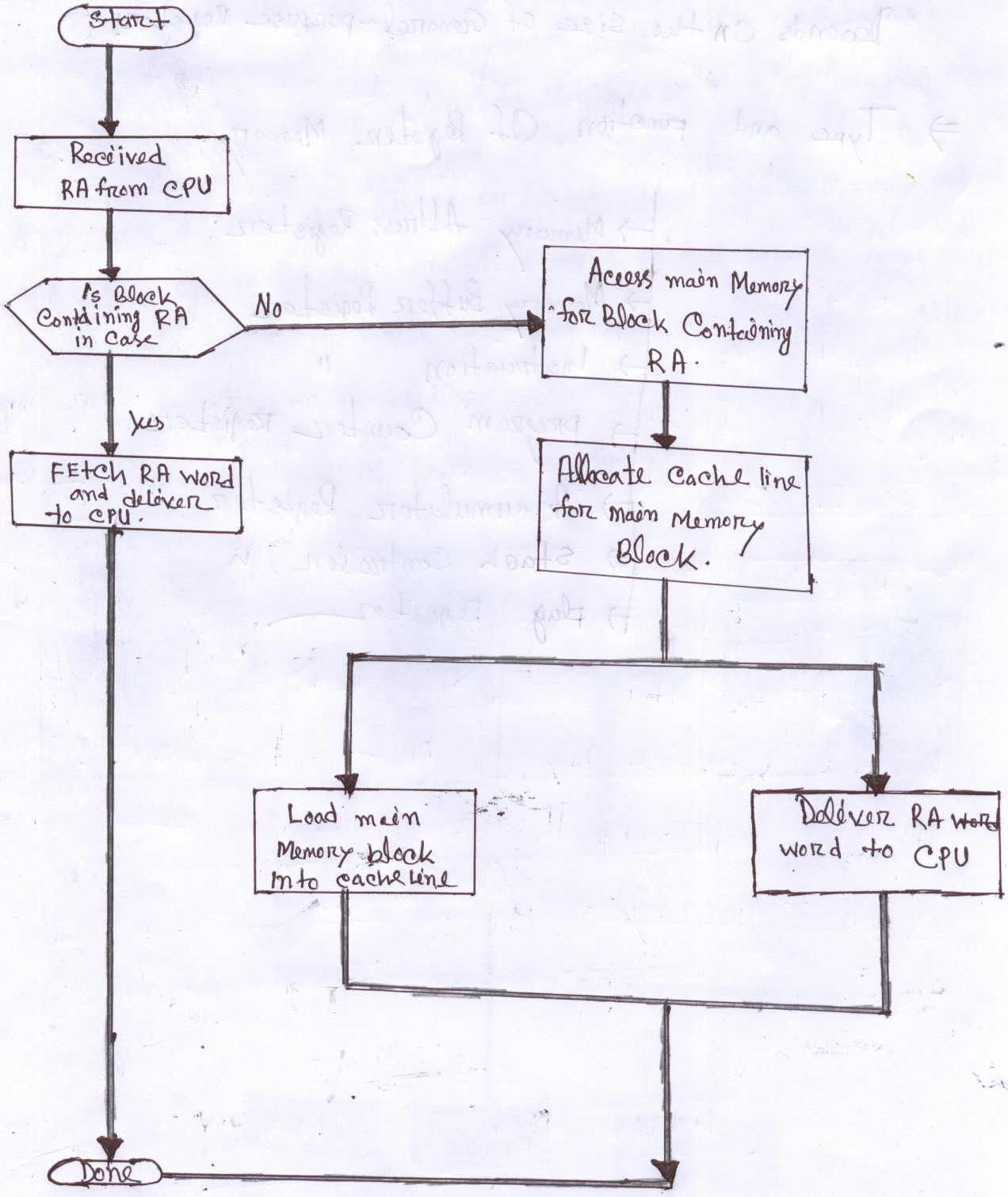


Figure: Cache Read Operation.