B. Tech CSE/IT

Nov-2019

Solution of COALP(ACCS-16303)

CSE-3rd Semester

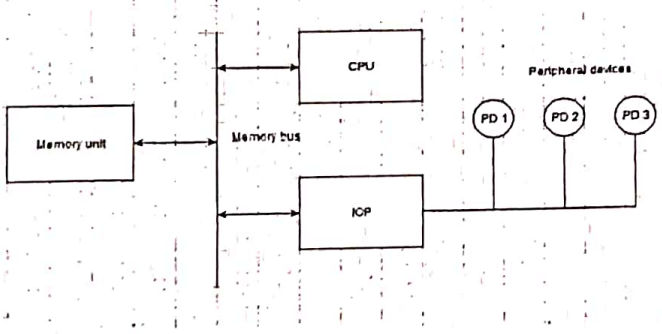
1.

i.

CISC	RISC
Emphasis on hardware	Emphasis on software
Multiple instruction sizes and formats	Instructions of same set with few formats
Less registers	Uses more registers
More addressing modes	Fewer addressing modes
Extensive use of microprogramming	Complexity in compiler
Instructions take a varying amount of cycle time	Instructions take one cycle time
Pipelining is difficult	Pipelining is easy

ii.

An **input-output processor** (IOP) is a processor with direct memory access capability. In this, the **computer** system is divided into a memory unit and number of processors. ... The **IOP** is similar to CPU except that it handles only the details of I/O processing. The **IOP** can fetch and execute its own instructions.



iii.

TABLE 5-4 Memory-Reference Instructions

Symbol	Operation decoder	Symbolic description
AND	D_0	$AC \leftarrow AC \wedge M[AR]$
ADD	D_1	$AC \leftarrow AC + M[AR], \ E \leftarrow C_{out}$
LDA	D_2	$AC \leftarrow M[AR]$
STA	D_3	$M[AR] \leftarrow AC$
BUN	D_4	$PC \leftarrow AR$
BSA	D_5	$M[AR] \leftarrow PC, \ PC \leftarrow AR + 1$
ISZ	D_6	$M[AR] \leftarrow M[AR] + 1,$ If $M[AR] + 1 = 0$ then $PC \leftarrow PC + 1$

iv.

Content-addressable memory is a special type of computer memory used in certain very-high-speed searching applications. It is also known as associative memory or associative storage and

compares input search data against a table of stored data, and returns the address of
data.

v.

11- address lines, 32- data lines

vi.

The main difference between hardware and software interrupt is that a ha
interrupt is generated by an external device while a software interrupt is generate
executing program. ... However, most modern computers can handle interrupts faste

vii.

Reg. Symbol	No. of Bits	Register	Register Function
DR	16	Data Register	Hold operand
AR	12	Address Register	Hold address for Mem
AC	16	Accumulator	Processor register
IR	16	Instruction Register	Hold instruction code
PC	12	Program Counter	Hold address of instruct
TR	16	Temp Register	Hold temp data
INPR	8	Input Register	Holds input char
OUTR	8	Output Register	Holds output char

viii.

HARDWIRED CONTROL UNIT
VERSUS
MICROPROGRAMMED
CONTROL UNIT

HARDWIRED CONTROL UNIT	MICROPROGRAMMED CONTROL UNIT
A unit that uses combinational logic units, featuring a finite number of gates that can generate specific results based on the instructions that were used to invoke those responses	A unit that contains microinstructions in the control memory to produc control signals
Speed of operations is fast	Speed of operations is slow because it requires frequent memory accesses
To do modifications, the entire unit should be redesigned	Modifications can be implemented by changing th microinstructions in the control memory
More costly to implement	Less costly to implement
It is difficult to handle complex instructions	It is easier to handle complex instructions
It is difficult to perform instruction decoding	Less difficult to perform instruction decoding
Uses a small instruction set	Ues a large instruction set
There is no control memory usage	Uses control memory
Used in processors that use a simple instruction set known as the Reduced Instruction Set Computers (RISC)	Used in processors based on complex instruction set know as Complex Instruction Set Computer (CISC)

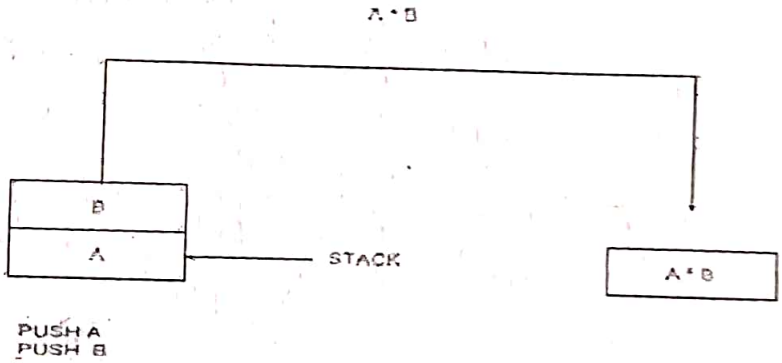
ix.

x.

Section-B

2.

1. **Zero Address Instructions –**



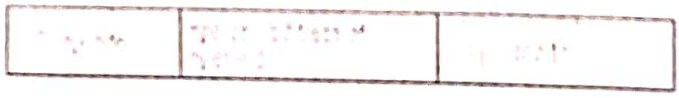
A stack based computer do not use address field in instruction.To evaluate a expression first it is converted to revere Polish Notation i.e. Post fix Notation.

```
Expression: X = (A+B)*(C+D)
Postfixed : X = AB+CD+*
TOP means top of stack
M[X] is any memory location
```

PUSH	A	TOP = A
PUSH	B	TOP = B

ADD		TOP = A+B
PUSH	C	TOP = C
PUSH	D	TOP = D
ADD		TOP = C+D
MUL		TOP = (C+D)*(A+B)
POP	X	M[X] = TOP

2. **One Address Instructions –**
This use a implied ACCUMULATOR register for data manipulation.One operand is in accumulator and other is in register or memory location.Implied means that the CPU already know that one operand is in accumulator so there is no need to specify it.

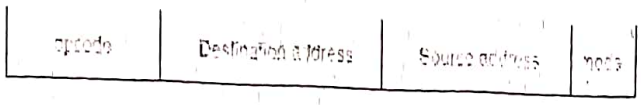


```
Expression: X = (A+B)*(C+D)
AC is accumulator
M[] is any memory location
M[T] is temporary location
```

LOAD	A	AC = M[A]
ADD	B	AC = AC + M[B]
STORE	T	M[T] = AC
LOAD	C	AC = M[C]
ADD	D	AC = AC + M[D]
MUL	T	AC = AC * M[T]

STORE	X	M[X] = AC

3. **Two Address Instructions –**
This is common in commercial computers.Here two address can be specified in the instruction.Unlike earlier in one address instruction the result was stored in accumulator here result cab be stored at different location rather than just accumulator, but require more number of bit to represent address.

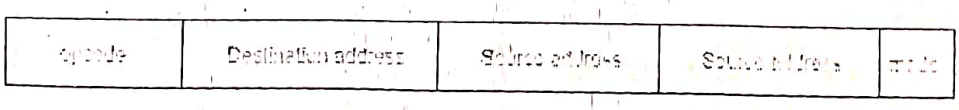


Here destination address can also contain operand.

```
Expression: X = (A+B)*(C+D)
R1, R2 are registers
M[] is any memory location
```

MOV	R1, A	R1 = M[A]
ADD	R1, B	R1 = R1 + M[B]
MOV	R2, C	R2 = C
ADD	R2, D	R2 = R2 + D
MUL	R1, R2	R1 = R1 * R2
MOV	X, R1	M[X] = R1

4. **Three Address Instructions –**
This has three address field to specify a register or a memory location. Program created are much short in size but number of bits per instruction increase. These instructions make creation of program much easier but it does not mean that program will run much faster because now instruction only contain more information but each micro operation (changing content of register, loading address in address bus etc.) will be performed in one cycle only.




```
Expression: X = (A+B)*(C+D)
```

R1, R2 are registers
M[ ] is any memory location

ADD	R1, A, B	R1 ← M[A] + M[B]
ADD	R2, C, D	R2 ← M[C] + M[D]
MUL	X, R1, R2	M[X] ← R1 * R2

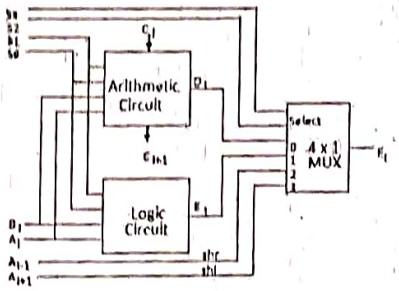
3.

Arithmetic Logic Shift Unit:

- The arithmetic logic unit (ALU) is a common operational unit connected to a number of storage registers
- To perform a microoperation, the contents of specified registers are placed in the inputs of the ALU
- The ALU performs an operation and the result is then transferred to a destination register
- The ALU is a combinational circuit so that the entire register transfer operation from the source registers through the ALU and into the destination register can be performed during one clock pulse period.

Shift Microoperations

ARITHMETIC LOGIC SHIFT UNIT



S3	S2	S1	S0	Cin	Operation	Function
0	0	0	0	0	F = A	Transfer A
0	0	0	0	1	F = A + 1	Increment A
0	0	0	1	0	F = A + B	Addition
0	0	0	1	1	F = A + B + 1	Add with carry
0	0	1	0	0	F = A + B'	Subtract with borrow
0	0	1	0	1	F = A + B' + 1	Subtraction
0	0	1	1	0	F = A - 1	Decrement A
0	0	1	1	1	F = A	Transfer A
0	1	0	0	X	F = A ∧ B	AND
0	1	0	1	X	F = A ∨ B	OR
0	1	1	0	X	F = A ⊕ B	XOR
0	1	1	1	X	F = A'	Complement A
1	0	X	X	X	F = shr A	Shift right A into F
1	1	X	X	X	F = shl A	Shift left A into F

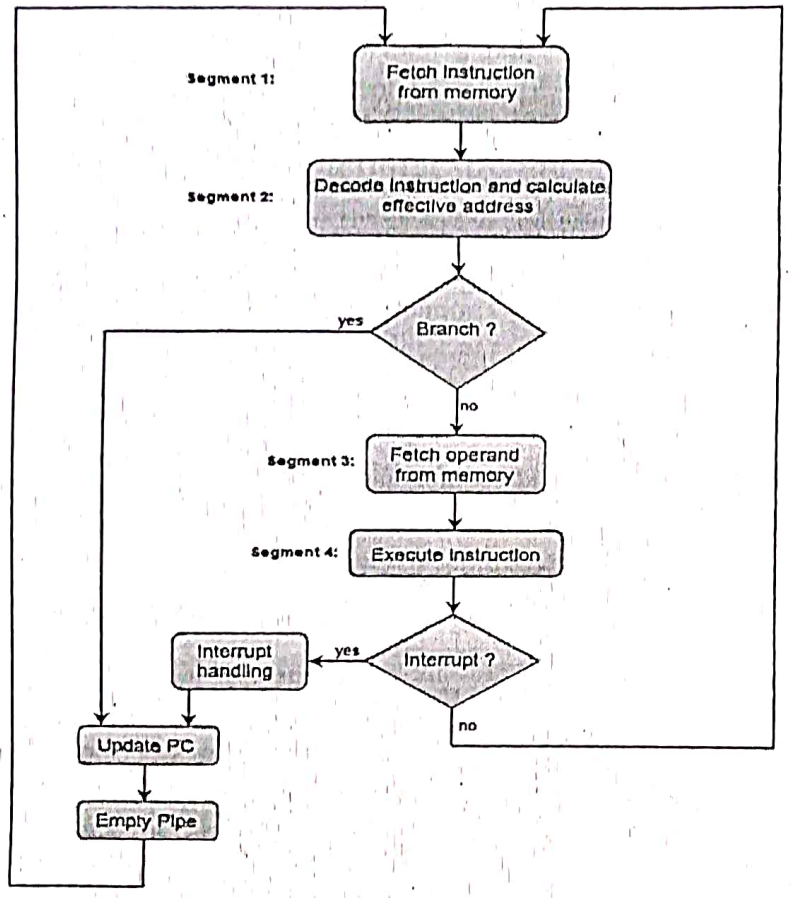
4.

A **Pipelining** is a series of stages, where some work is done at each stage in parallel. The stages are connected one to the next to form a pipe - **Instructions** enter at one end, progress through the stages, and exit at the other end.

Instruction Pipeline

- **Instruction execution process lends itself naturally to pipelining**
  - **overlap the subtasks of instruction fetch, decode and execute**
- **Instruction pipeline has six operations,**
  - Fetch Instruction (FI)
  - Decode Instruction (DI)
  - Calculate operands (CO)
  - Fetch operands (FO)
  - Execute Instructions (EI)
  - Write result (WR)

Overlap these operations



5.

4-18

(a) A = 11011001
B = 10110100 (⊕)
A ← A⊕B 01101101

A = 11011001
B = 11111101 (OR)
11111101 A ← A∨B

4-19

(a) AR = 11110010
BR = 11111111 (+)
AR = 11110001 BR = 11111111 CR = 10111001 DR = 11010101

(b) CR = 10111001
DR = 11101010 (AND)
CR = 10101000 BR = 11111111 + 1
BR = 00000000 AR = 11110001 DR = 11101010

(c) AR = 11110001
CR = 10101000 (−)
AR = 01001001 ; BR = 00000000 ; CR = 10101000 ; DR = 11101010

6. The memory unit of a computer has 256K words of 32 bits each. The computer has an instruction format with four fields: an operation code field, a mode field to specify one of the seven addressing modes, a register address field to specify one of the 60 processor registers, and a memory address. Specify the instruction format and the number of bits in each field if the instruction is in one memory word.

Answer: Total (Word) = 32 bits Mode Field: 7 Addressing Modes: Requires 3 bits (23=8≈7)Register Address Field: 60 Registers: Requires 6 bits (26= 64≈60)256 K = 28x 210= 218and can be addressed with 18 bits Opcode: 32 – (18+6+3) = 32 – 27 = 5 bits.

Section –C

7.

Types of Addressing Modes

Below we have discussed different types of addressing modes one by one:

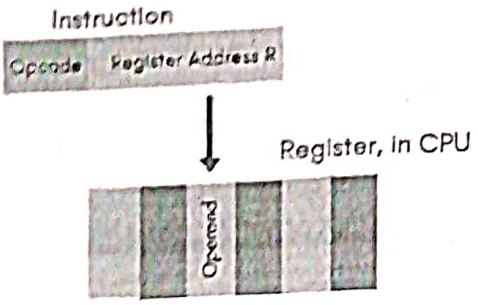
Immediate Mode

In this mode, the operand is specified in the instruction itself. An immediate mode instruction has an operand field rather than the address field.

For example: ADD 7, which says Add 7 to contents of accumulator. 7 is the operand here.

Register Mode

In this mode the operand is stored in the register and this register is present in CPU. The instruction has the address of the Register where the operand is stored.



dvantages

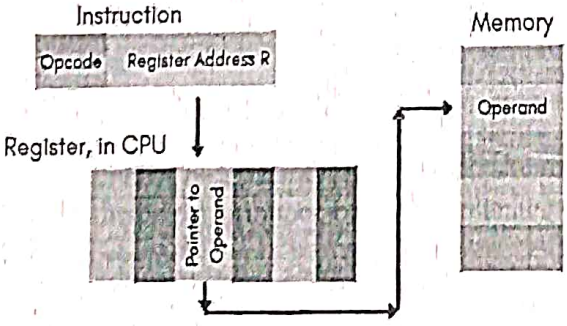
- Shorter instructions and faster instruction fetch.
- Faster memory access to the operand(s)

)isadvantages

- Very limited address space
- Using multiple registers helps performance but it complicates the instructions.

Register Indirect Mode

n this mode, the instruction specifies the register whose contents give us the address of perand which is in memory. Thus, the register contains the address of operand rather than he operand itself.



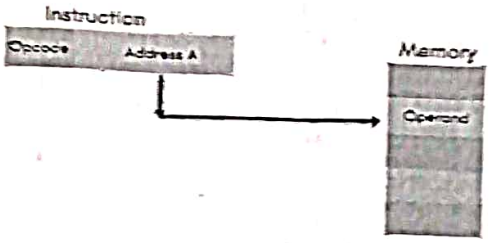
Auto Increment/Decrement Mode

n this the register is incremented or decremented after or before its value is used.

Direct Addressing Mode

In this mode, effective address of operand is present in instruction itself.

- Single memory reference to access data.
- No additional calculations to find the effective address of the operand.

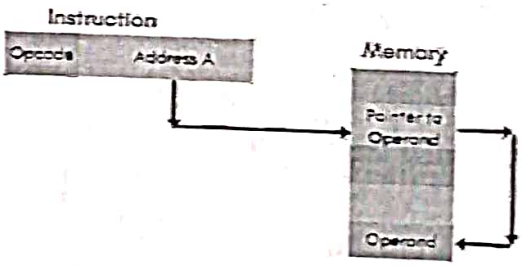


For Example: ADD R1, 4000 - In this the 4000 is effective address of operand.

NOTE: Effective Address is the location where operand is present.

Indirect Addressing Mode

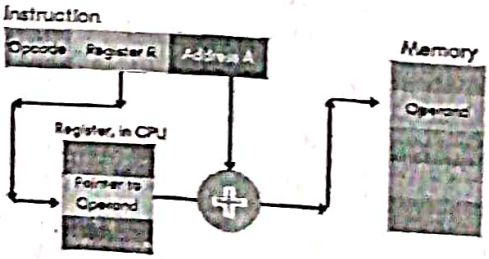
In this, the address field of instruction gives the address where the effective address is store in memory. This slows down the execution, as this includes multiple memory lookups to fi the operand.



Displacement Addressing Mode

In this the contents of the indexed register is added to the Address part of the instruction, to obtain the effective address of operand.

EA = A + (R), In this the address field holds two values, A(which is the base value) and R(that holds the displacement), or vice versa.



Relative Addressing Mode

It is a version of Displacement addressing mode.

In this the contents of PC (Program Counter) is added to address part of instruction to obtain the effective address.

EA = A + (PC), where EA is effective address and PC is program counter.

The operand is A cells away from the current cell(the one pointed to by PC)

Base Register Addressing Mode

It is again a version of Displacement addressing mode. This can be defined as EA = A + (R), where A is displacement and R holds pointer to base address.

8.

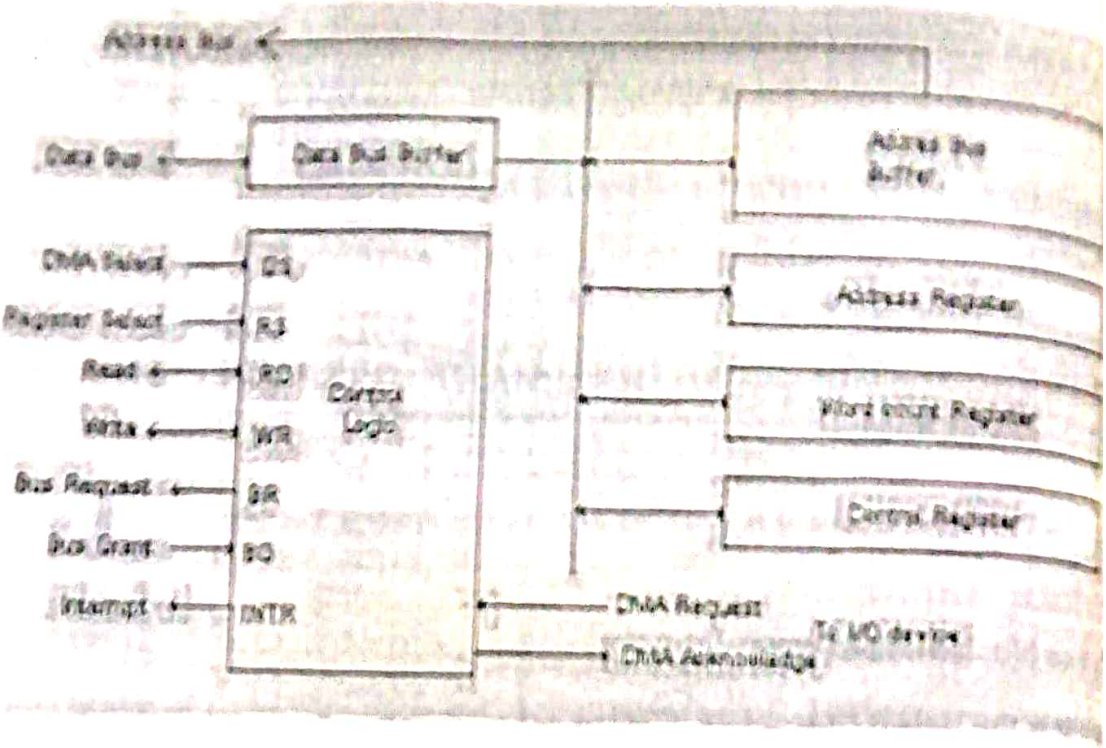
Direct memory access (DMA) is a feature of computer systems that allows certain hardware subsystems to access main system memory (random-access memory), independent of the central processing unit (CPU).

Burst or block transfer DMA

- It is the fastest DMA mode. In this two or more data bytes are transferred continuously.
- Processor is disconnected from system bus during DMA transfer. N number of machine cycles are adopted into the machine cycles of the processor where N is the number of bytes to be transferred.
- DMA sends HOLD signal to processor to request for system bus and waits for HLDA signal.
- After receiving HLDA signal, DMA gains control of system bus and transfers one byte. After transferring one byte, it increments memory address, decrements counter and transfers next byte.
- In this way, it transfer all data bytes between memory and I/O devices. After transferring all data bytes, the DMA controller disables HOLD signal & enters into slave mode.

2) Cycle steal or single byte transfer DMA.

- In this mode only one byte is transferred at a time. This is slower than burst DMA.
- DMA sends HOLD signal to processor and waits for HLDA signal on receiving HLDA signal, it gains control of system bus and executes only one DMA cycle.
- After transfer one byte, it disables HOLD signal and enters into slave mode.
- Processor gains control of system bus and executes next machine cycle. If count is not zero and data is available then the DMA controller sends HOLD signal to the processor and transfer next byte of data block.

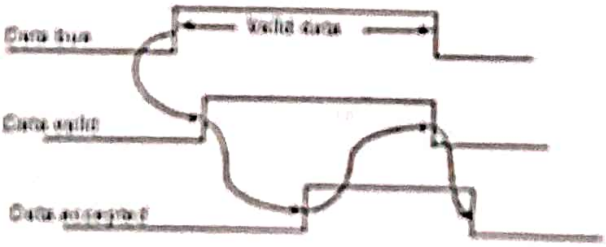


9.

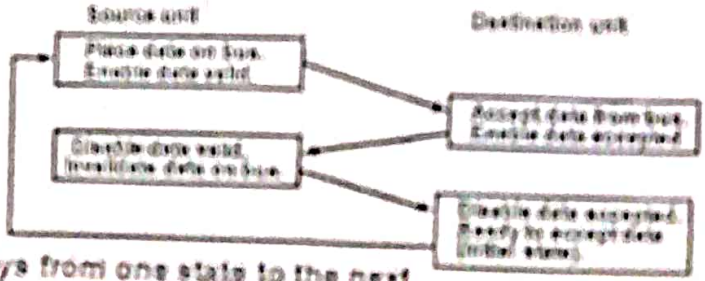
SOURCE-INITIATED TRANSFER USING HANDSHAKE

Block Diagram

Timing Diagram



Sequence of Events

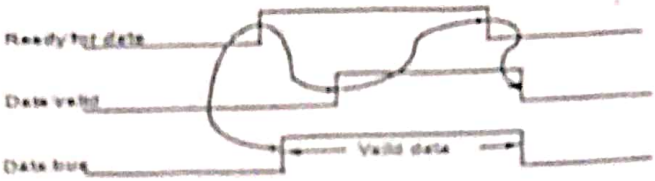


- Allows arbitrary delays from one state to the next
- Permits each unit to respond at its own data transfer rate
- The rate of transfer is determined by the slower unit

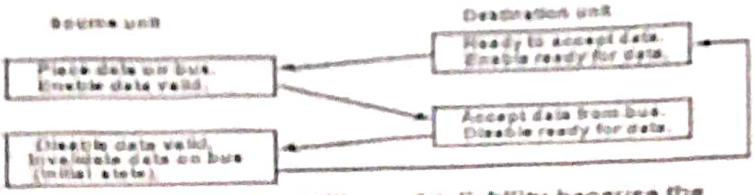
DESTINATION-INITIATED TRANSFER USING HANDSHAKE

Block Diagram

Timing Diagram



Sequence of Events



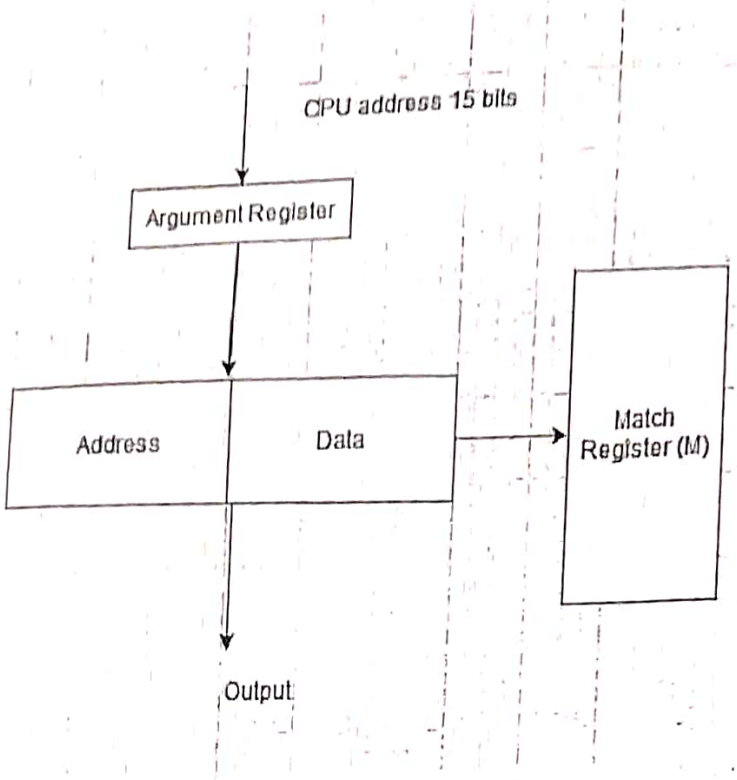
- Handshaking provides a high degree of flexibility and reliability because the successful completion of a data transfer relies on active participation by both units
- If one unit is faulty, data transfer will not be completed
  -> Can be detected by means of a *timeout* mechanism

he transformation of data from main memory to cache memory is called mapping.
here are 3 main types of mapping:

- Associative Mapping
- Direct Mapping
- Set Associative Mapping

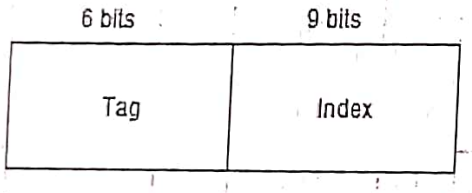
Associative Mapping

he associative memory stores both address and data. The address value of 15 bits
s 5 digit octal numbers and data is of 12 bits word in 4 digit octal number. A CPU
ddress of 15 bits is placed in **argument register** and the associative memory is
earched for matching address.



Direct Mapping

The CPU address of 15 bits is divided into 2 fields. In this the 9 least significant bi constitute the **index** field and the remaining 6 bits constitute the **tag** field. The number of bits in index field is equal to the number of address bits required to acc cache memory.



Set Associative Mapping

The disadvantage of direct mapping is that two words with same index address can reside in cache memory at the same time. This problem can be overcome by set associative mapping.

this we can store two or more words of memory under the same index address.
ach data word is stored together with its tag and this forms a set.