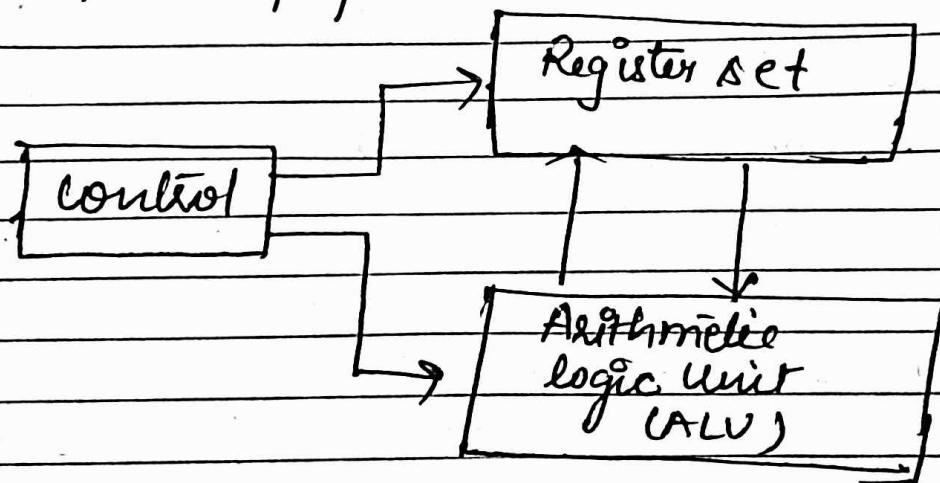


Central Processing Unit

- The part of the computer that performs the bulk of data processing operations.
- The register set stores intermediate data used during the executions of the instructions.
- The ALU performs the required microoperations for executing the instructions.
- The control unit (CU) supervises the transfer of information among the registers & instructs the ALU as to which operation to perform.



Major Components of CPU

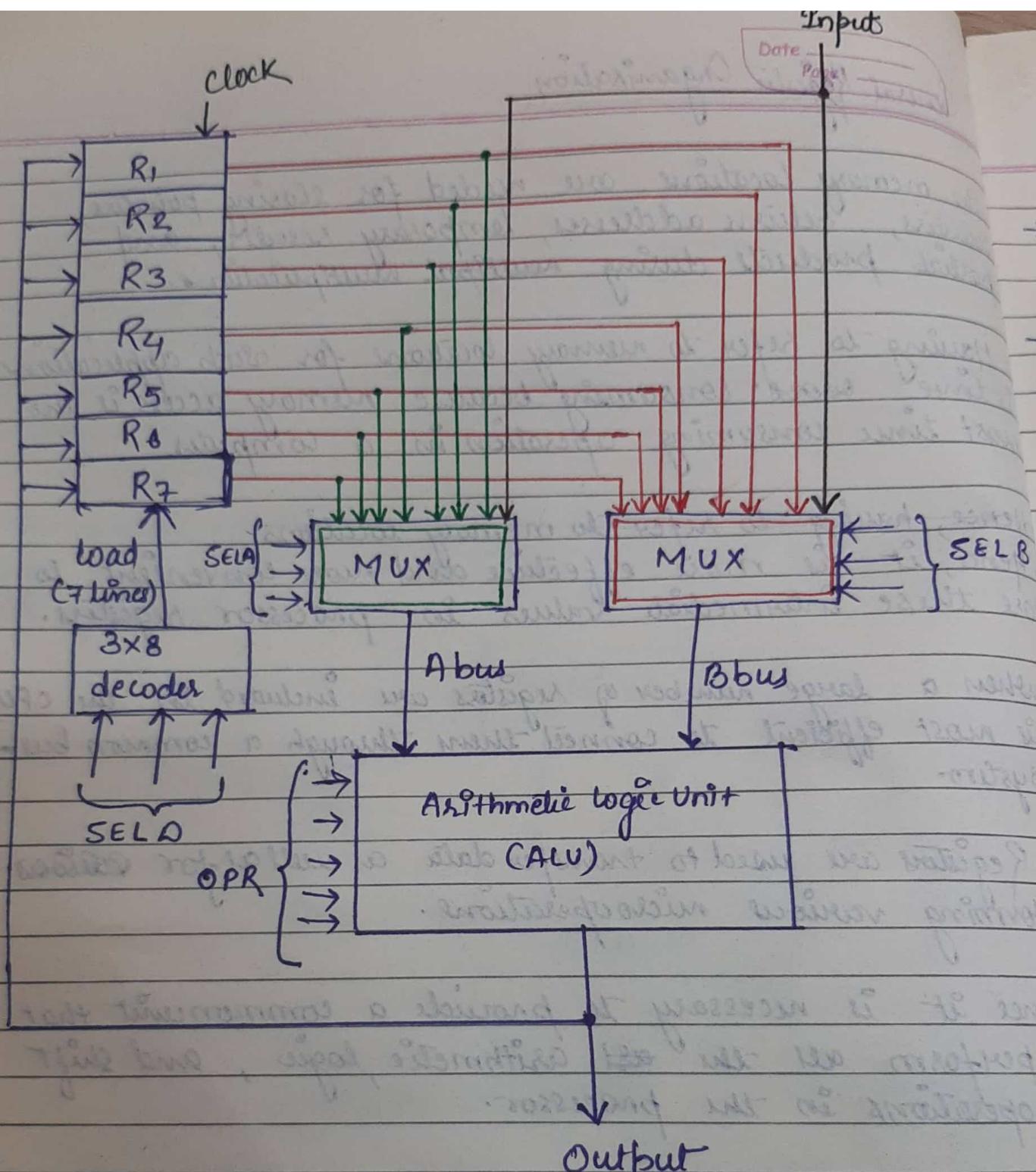
- Storage components :- registers, flip flops
- Execution components :- ALU
- Transfer components : BUS
- Control components : Control unit

② General Register Organisation

Date _____

Page _____

- The memory locations are needed for storing pointers, counters, return addresses, temporary results, and partial products during multiple multiplications.
- Having to refer to memory locations for such applications is time consuming because memory access is the most time consuming operation in a computer.
- Hence, having to refer to memory locations
- Hence, it is more effective and more convenient to store these intermediate values in processor registers.
- When a large number of registers are included in the CPU it is most efficient to connect them through a common bus system.
- Registers are used to transfer data as well as for performing various microoperations.
- Hence it is necessary to provide a common unit that can perform all the arithmetic, logic, and shift microoperations in the processor.
- Therefore, A bus organisation for Seven CPU register is shown in the fig.



(a) Block diagram

3	3	3	5
SEL A	SEL B	SELD	OPR

(b) control word

REGISTER SET WITH COMMON ALU

Diagram Description

Date _____
Page _____

Multiplexers:-

- The output of each register is connected to two multiplexers (MUX) to form two buses A and B.
- The Selection lines in each multiplexer select one register or input data for the particular bus.

Arithmetic logic Unit (ALU)

- The A and B buses form the inputs to a common arithmetic logic unit (ALU).
- The operation selected in the ALU determines the arithmetic or logic microoperation that is to be performed.
- The result of the microoperation is available for output data and also goes into the inputs of all the registers.

Decoders

- The register that receives the information from the output bus is selected by a decoder.
- The decoder activates one of the register loads inputs, thus providing a transfer path between the data in the output bus and the inputs of the selected destination register.

Control unit:- The control unit that operates the CPU bus system directs the information flow through the registers and ALU by selecting the various components in the system.

Example

Example which perform the operation given below using the above diagram

$$R_1 \leftarrow R_2 + R_3$$

→ The control must provide binary selection variables to the following selector inputs:

1. Mux A Selector (SEL A): 001 to place the content of R_2 into bus A.

2. Mux B Selector (SEL B): 010 to place the content of R_3 into bus B.

3. ALU Operation Selector (OPR): 00010 to provide the arithmetic operation addition $A+B$.

4. Decoder Destination Selector (SEL D): 011 to transfer the content of the output bus into R_1 .

CONTROL WORD

- The combine word value of a binary selection inputs specifies the control value word.
- Control words for all micro operation are stored in the control memory.
- It consists of four fields SELA, SELB and SELD contains three bits each and OPR field contains five bits.
- Therefore, the total bits in the control word are 14.

3	3	3	5			
SELA	SELB	SELD	OPR			

FORMAT OF CONTROL WORD

- 3-bits of SELA select a source register of the A input
- 3-bits of SELB select a source register of the B input
- 3-bits of SELD or SELREG select a destination register using the decoder.
- 5-bits of OPR select the operation to be performed by ALU.

Encoding of Register Selection field.

Binary code	SELA Input	SELB Input	SELQ
000			None
001	R ₁	R ₁	R ₁
010	R ₂	R ₂	R ₂
011	R ₃	R ₃	R ₃
100	R ₄	R ₄	R ₄
101	R ₅	R ₅	R ₅
110	R ₆	R ₆	R ₆
111	R ₇	R ₇	R ₇

SELA \rightarrow R₁ \rightarrow 001

SELB \rightarrow R₂ \rightarrow 010

SELQ \rightarrow R₃ \rightarrow 011

Encoding of ALU operations

S.No.	OPR Select	Operation	Symbol
1	00000	Transfer A	TSA
2	00001	Invert A	INA
3	00010	Add A+B	ADD
4	00101	Subtract A-B	SUB
5	00110	Decrement A	DECA
6	01000	ADD A and B	AND
7	01010	OR A and B	OR
8	01100	XOR A and B	XOR
9	01110	Complement A	COMA
10	10000	Shift right A left	SHRA
11	11000	Shift left A	SHL A

ALU Micro-Operations

Date _____
Page _____

Micr-operation	SEL A	SEL B	SEL C	OPR.	Control word			
$R_1 \leftarrow R_2 - R_3$	R_2	R_3	R_1	SUB	010	011	001	1010101
$R_4 \leftarrow R_4 \vee R_5$	R_4	R_5	R_4	OR	100	101	100	01010
$R_6 \leftarrow R_6 + 1$	R_6	-	R_1	INCA	110	000	110	00001
$R_7 \leftarrow R_1$	R_1	-	R_7	TSFA	001	000	111	00000
Output $\leftarrow R_2$	R_2	-	None	TSFA	010	000	000	000000
Output \leftarrow Input	Input	-	None	TSFA	000	000	000	00000
$R_4 \leftarrow Shl R_4$	R_4	-	R_4	SHLA	100	000	100	11000
$R_5 \leftarrow 0$	R_5	R_5	R_5	XOR	101	101	101	01100

Advantages of General Registers

- ① Efficiency of CPU increases because large number of registers are used.
- ② Less memory space is required because here the instructions are written in a very compact way.

Disadvantages

- ① Cost will be high because of large number of registers.
- ② Also you have to make sure of unnecessary use of registers.

Stack Organization

- A stack is a storage device that stores information in a Last-In-First-Out (LIFO) manner.
- The stack in digital computers is essentially a memory unit with an address register (AR) that can count only (after an initial value is loaded into it).
- The register that holds the address for that stack is called a stack pointer (SP) because its values always points at the top item in the stack.

Two operations of stack

- ① The insertion and deletion of items.
The operation of insertion is called push (or push-down) because it can be thought of as the result of pushing a new item on top.
- ② The operation of deletion is called pop (or pop-up) because it can be thought of as the result of removing one item so that the stack pops up.

Note:-

However, nothing nothing is pushed or popped in computer stack pointer register. These operations are simulated by incrementing or decrementing the stack pointer register.

Implementation of Stack

In digital computers, stack can be implemented in two ways:

- Register stack
- Memory stack

REGISTER STACK

A stack can be organized as a collection of finite number of registers that are used to store temporary information during the execution of a program.

Figure shows the organization of a 64-word register stack. It consists of 4 variables which can store different type of values which are as follows

① Stack pointer register: Denoted by SP and it contains a binary number whose value is equal to the address of the word.
→ In a 64 word stack, the stack pointer contains 6 bit because $2^6 = 64$. Since SP has only 6 bits, it cannot exceed a number greater than 63 (111111 in binary).

② FULL :- It is a bit register, set to 1 when the Stack is full.

③ Empty :- It is a bit register, Set to 1 when the Stack is Empty.

④ Data Register (DR) :- It is a data register that holds the binary data to be written into or read out of the stack.

FULL

EMPTY

63

SP →

C

B

A

D

DR

4

3

2

1

0

BLOCK DIAGRAM OF 64-WORD STACK

→ Three-element A, B, and C are located in the stack arrangement.

→ The stack pointer register (SP) holds the address of the element present at the top of the stack.

→ Element C is at the top of the stack and SP holds the address of C i.e., 3.

- The top element is popped from the stack through reading memory word at address 3 and decrementing the SP by 1.
 - Then, B is at the top of the stack and the SP holds the address of B that is 2.
 - It can insert a new word, the stack is pushed by incrementing the stack pointer by 1 word and inserting a word in that incremented location.
- Initially, SP is 0, EMPTY is 1, and FULL is 0
 So that, SP points to the word address 0 and stack is marked Empty.
- So if the stack is not full then push operation is implemented and the following microoperation is performed.

$SP \leftarrow SP + 1$

Increment stack pointer

$M[SP] \leftarrow DR$

Write item on top of the stack.

0 of 38 of PGM3

IV Case Note :-

- ① SP holds address of the top of the stack and
- ② M[SP] denotes the memory word specified by the address presently available in SP.
- ③ The first item is stored at in the stack at address 1. and The last item is stored in the stack at address 0.

II Case :-

If SP reaches 0 means stack is full of items
So FULL is set to 1

This condition is reached if the top item prior to the last push was in location 63 and, after incrementing SP, the last item is stored in location 0.

Once an item is stored in location 0, there are no more empty registers in the stack.

So EMPTY is set to 0

If $CSP = 0$ then ($FULL \leftarrow 1$) Check if stack is full
 $EMTY \leftarrow 0$ Mark the stack not empty.

POP Operation :-

- A item is deleted from the stack if the stack is not empty (if $EMTY = 0$).
→ The pop operation consists of the following sequence of micro operations:

$DR \leftarrow M[SP]$ Read item from the top of stack

$SP \leftarrow SP - 1$ Decrement stack pointer

If $(SP = 0)$ then ($EMTY \leftarrow 1$) check if stack is empty

$FULL \leftarrow 0$ Mark the stack not full

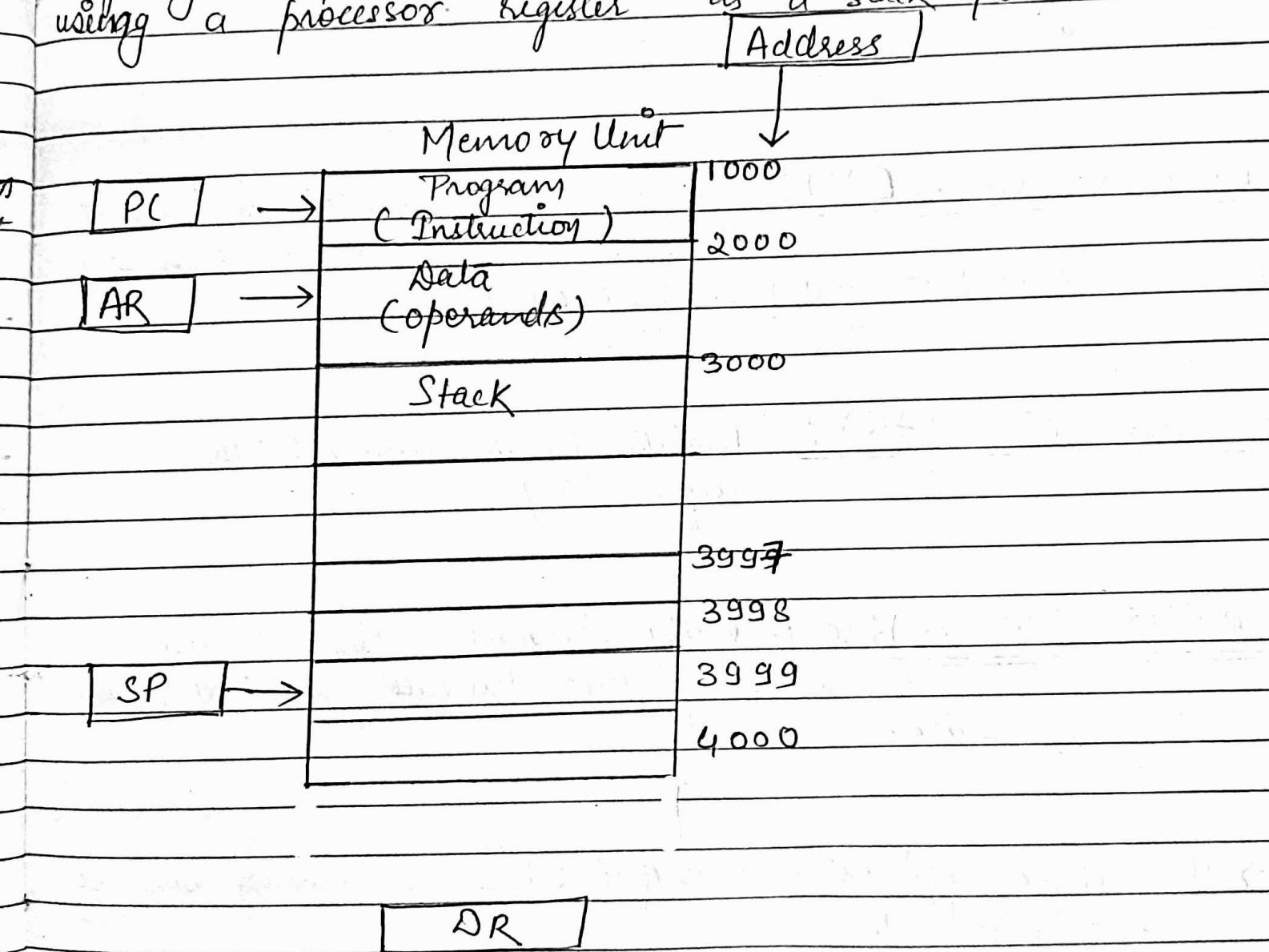
Memory Stack

Page _____

→ A stack can exist as a stand alone unit.

→ It can be implemented in a random access memory (RAM) attached to a CPU

→ The implementation of a stack in the CPU is done by assigning a portion of memory to a stack operation and using a processor register as a stack pointer.



→ Computer Memory with Program, data and stack segments.

→ An area of the computer memory is broken into three segments such as:

- Program

- Data

- Stack

Program Counter (PC): The address of the next instruction in the program is saved in the pointer called program counter (PC).

Address Register (AR): points to an array of the information

SP (Stack Pointer): continually influences the address of the elements present at the top of the stack.

→ The three registers that are linked to the common bus are PC, AR, and SP.

→ An operand is read during execute stage using the address register.

- SP points to a beginning value '4001' & increases with decreasing addresses
- The first element is saved at address 4000, the next element is saved at address 3999 and last element at address 1000

PUSH OPERATION

- The data register can read an element into or form the stack. It can use push operation to insert a new element into the stack.

$SP \leftarrow SP - 1$
 $M[SP] \leftarrow DR$

POP OPERATION

It can delete an element from the stack. It can use the pop operation which is as follows

$DR \leftarrow M[SP]$
 $SP \leftarrow SP + 1$

Date _____
Page _____

- Two processor registers are used to check the stack limits
- One processor register for the upper limit (1000) and
- one for lower limit (400)
- During push operation, the SP is compared with the upper limit to check if the stack is full.
- During pop operation, the SP is compared with the lower limit to check if the stack is empty.