

TOPIC NAME:

Law
1st Class
16.10.2024

Law 2211

- Cyber crime
- intellectual arts

• wingless software

• cyber security Law 2023

• 15 days 100 problems

Algorithm
1st class (17.10.2024)

Implementation, bruteforce
adverb, 15 days 100 problems

15 days 100 problems

Architecture
1st class (10.10.2024)

structure function

Computer Organization and
Architecture

Designing for Performance
eighth edition

william stallings

1.1. Organization and Architecture

- visible attributes during programming its an architecture
↳ not-changeable

1. Interaction (char)

2. datatype

3. I/O mechanism

↳ mouse, keyboard, harddisk

4. Memory addressing technique

- visible hardware units
to programmers its an organization. (Page 32 pdf)
↳ changeable

hardware units changeable but
software units non-changeable

basic hardware → adder

→ subtraction

Anch: Multi

org: hardware multi

→ multiplication
→ division

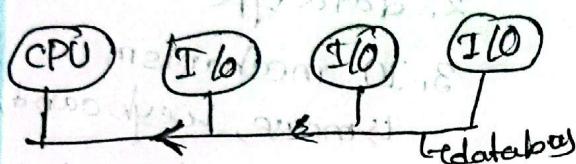
main unit for computer

7. Control, memory, I/O, Arithmetic
units

1.2 Structure and function

- A functional view of computer

- data communication



- control units work like a traffic controller

- Possible types of computer operation.

- structure of the computer

→ within system peripheral

→ outin system data communication

- Top to Bottom approach

1. database
 2. Address bus
 3. Control bus
- System bus
- 3 types of bus

load → read
store → write

- internal memory in CPU is called register.

met

Computer Architecture: refers to those attributes of a system visible to a programmer or put another way, those attributes that have a direct impact on the logical execution of a program.

Computer Organization: refers to the operational units and their interconnections that realize the architectural specifications.

- Structure: The way in which the components are interconnected

- function: The operation of each individual component as part of the structure

TOPIC NAME: _____

DAY: _____

TIME: _____

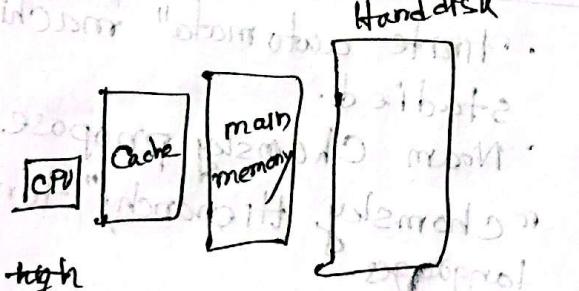
DATE: / /

22.10.2024

Architecture

- Multicore computer structure
- An individual processing unit unit on processor chip
- simplified view of Major Elements of a Multicore

Computer



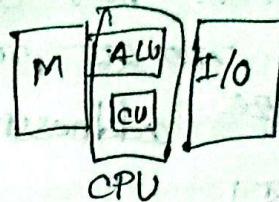
speed!

CPU > Cache > main memory > Hard disk

Chapter 2 (A brief history)

- Computer History
- 1st generation computer
- Von Neumann machine
- IAS computer Structure

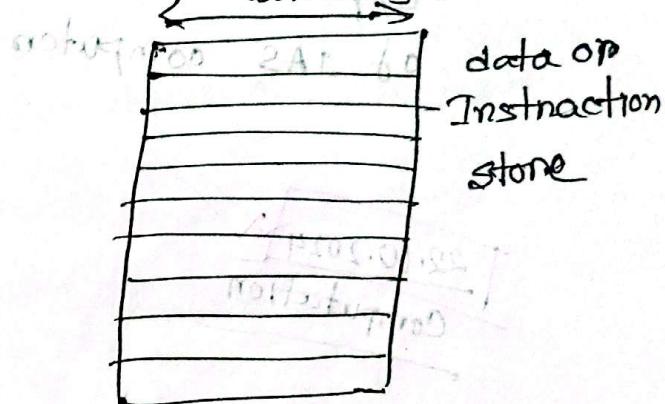
GOOD LUCK



1952

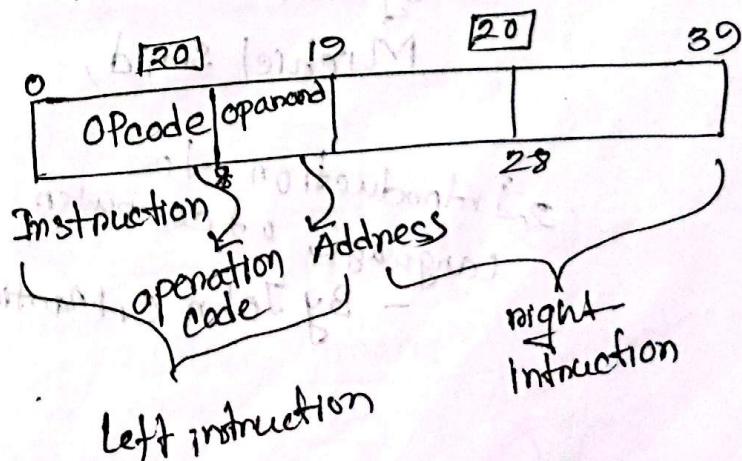
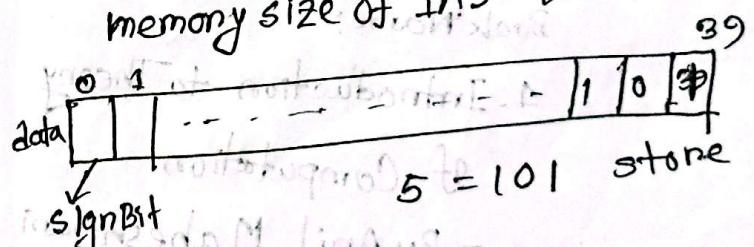
IAS computer

- IAS computer (1st generation)



Memory

memory size of IAS 40 bit



left instruction

(b) instruction word

$$2^{12} = 4096$$

TOPIC NAME :

Address bus ক্ষেত্র
size অতি বড়ো memory
2 size addressing এতে
পারবে।

Expanded structure
of IAS computer

TOPIC NAME :

DAY : _____

TIME : _____

DATE : / / /

27.10.2024

Archi
Architecture

A. Expanded Structure of IAS Computer.

• The IAS Instruction set

Table 2.1

• End Page 50

• An IBM 7094 configuration (2nd generation)

Figure 2.15

• Transistor

3rd generation IC

↳ status register

new as LSI

Fig: 1.17

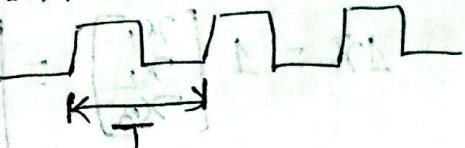
1st an 3rd generation configuration of LSI

(Exam) performance measure (p-49)

↳ Math

Crystal Ck,

Speed : Hz 66 GHz



$$T = \frac{1}{66 \times 10^9} \text{ s}$$

Clock

Processing

Working procedure of CPU
(P-60 pif) within one cycle

page 45 → math

Instruction Per Second (IPS)

$$IPS = \frac{N}{T}$$

$$\Rightarrow T = \frac{N}{IPS}$$

Cycle per Instruction (CPI)

$$CPI = \frac{f \times 10^6}{IPS}$$

Execution time

$$T = \frac{N \times CPI}{f \times 10^6}$$

TOPIC NAME: _____

DAY: _____

TIME: _____

DATE: / /

29.10.2024

Architecture

Exam

1. Structure \rightarrow IAS

page 412

Performance considerations:

- IBM 801 reduce instruction competition.
- Data and instruction movement time.

Exam

CPI, IPS, math

CP2 \rightarrow problem 1, 2
startings, page 75 pdf

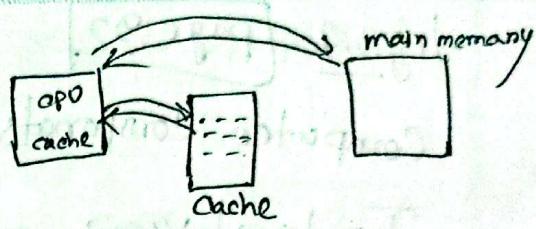
CP3 CPI, IPS, math
MIPS

2nd problem

Speedup techniques:

3 types:

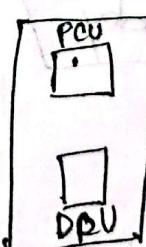
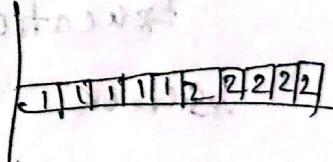
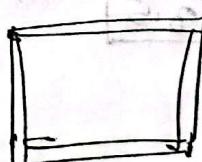
- ① Cache
- ② Pipe lining
- ③ Super-scale



• Cache size increasing

② Pipe lining

- ① Fetch
- ② decode
- ③ Load
- ④ Execute
- ⑤ Store



work alternative one
after another PCU and DPU

Exam parallel instruction pipe line

③ superscale

hardware & software
approach

TOPIC NAME :

Fig 3.2

page 92

• Computer Components

Top-level view

3.3 Basic instruction cycle.

3.4 Hypothetical machine

72 page — 95 page (pdf)

• Example of program

Execution 1.3.5

* See book

• Interrupts

next day

31.10.2024
Architecture

• Interrupts (বার্জিনেওয়া)

• Classes of Interrupts

- ↳ program

- ↳ Time

- ↳ I/O

- ↳ Hardware/firmware

(i) no, Interrupts,

(ii) sometimes "

(iii) always "

• Transfer of control via Interrupts

fig 3.9

Instruction Cycle with interrupts

• step single interrupt

• multiple "

(sequential) "

• nested "

④ 2nd approach (priority)

A) Interconnection Structures

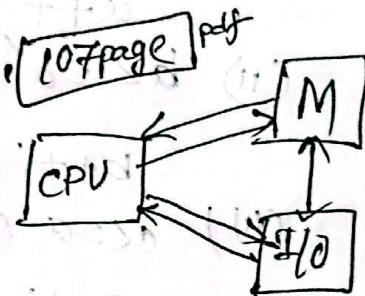
- memory

- I/O (interrupts)

- CPU

Fig 3.10 Computer module

• Bus Interconnection Scheme



(DMA)
• Direct Memory Access

• Control Line

31.10.2024

Law

- malware

- viruses

- Trojan

- worms

- phishing

• DDoS Attack

- Man-in-the-middle Attack

- Drive-by Download Attack

TOPIC NAME:

DAY:

TIME:

DATE:

CR

Source & destination register
Disktina use

Processor node
node as shortest
wanshall.

Delmenfor (negative value)

20C Disktina

shortest path I

03.11.2024
Architecture

traditional bus architecture

- local bus

- High performance architecture

- typical desktop system

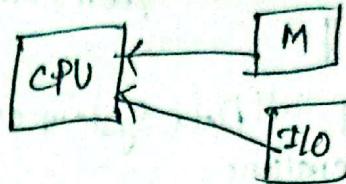
- PCI bus (high speed)

Chapter 4.0 - 3
processor Basic

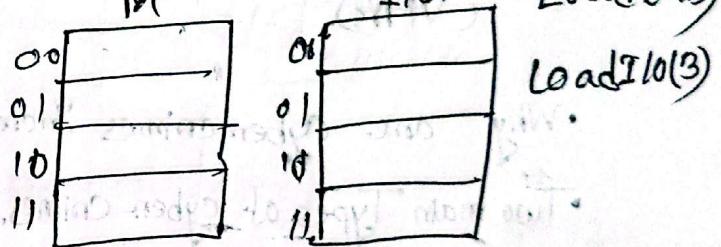
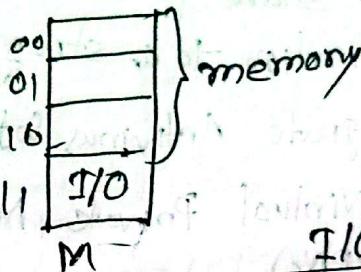
• fundamental of CPU

• processor - memory communication

• memory-mapped I/O



- Memory mapped I/O Load M(3)
- I/O mapped I/O



- User and supervisor mode

user mode interrupt supervisor mode

• CPU operation

• fig 3.3

• small accumulator base CPU

• Instruction general format

$$x_1 = f(x_2, x_3)$$

Ur M(A) On Register CPU

R₁

program control (PC)

$$IR, AR = M(PC)$$

opcode address

TOPIC NAME:

DAY:

TIME:

DATE: / /

Sum program

$$AC = M(X)$$

$$DR = AC$$

$$AC = M(Y)$$

$$AC = AC + DR$$

$$M(Z) = AC$$

$$AC = M(X)$$

$$AC = AC + M(X)$$

$$M(Z) = AC$$

computation data register

scam register

stack pointer

fig 8-7

A typical CPU with the general

$$AC = M(X)$$

$$DR = M(Y)$$

PCU

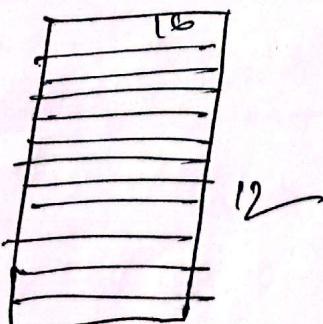
DPV: AC, DR

PCU: IR, AR, PC

page 147

Additional feature

Data
Address } multipurpose register
} (register files)



IBM 360 → 32 register

Additional data

, , , instruction

, , , scheme

TOPIC NAME:

DAY:

TIME:

DATE:

- square root recomposition
- how to apply this algorithm in graph

07.11.2024
Architecture

262 page:

$$x + y \rightarrow 2^s (y)$$
$$x - y$$
$$y \rightarrow \bar{y} + 1$$

$$x \oplus s = x\bar{s} + \bar{x}s$$

If $s=0$

$$x \oplus 0 = x \cdot 1 + \bar{x} \cdot 0 = x$$

If $s=1$

$$x \oplus 1 = x \cdot 0 + \bar{x} \cdot 1$$
$$= \bar{x}$$

9.4

An n-bit two's-complement adder-subtractor.

• overflow

• carry + overflow

$$\begin{array}{r} 53 \\ + 25 \\ \hline 78 \end{array}$$

$$\begin{array}{r} 0110101 \\ 0011001 \\ \hline 0100110 \end{array}$$

If unsigned $c=0$

If signed $v=0$

$$\begin{array}{r} 53 \\ - 41 \\ \hline 144 \end{array}$$

If signed $c=0$

$$\begin{array}{r} 00110101 \\ 11010011 \\ \hline 100001000 \end{array}$$

signed $v=0$

unsigned $c=1$

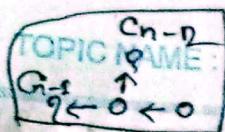
$$\begin{array}{r} 0011110 \\ - 1101001 \\ \hline 10110001 \end{array}$$

signed $v=1$
unsigned $c=1$

8 bit
unsigned: 0-255
signed: -128-127
carry: unsigned
overflow: signed

C = carry
v = overflow

$a_0, a_1, \dots, a_{n-1}, a_n$



$0, 1, \dots, n$

$$V = C_{n-1} \oplus C_{n-2}$$

if $V=1$

• if overflow done

if $V=0$

• if overflow not done

equation 9.6

- High speed adders:
- carry-look-ahead adder

$$c_i = x_i y_i + x_i c_{i-1} + y_i c_{i-1}$$

= $g_i p_i$

$$c_i = g_i + c_{i-1} p_i$$

$$c_{i-1} = g_{i-1} + p_{i-1} c_{i-2}$$

$$c_{i-1} = g_{i-1} + c_{i-2} p_{i-2}$$

$$c_i = g_i + p_i [g_{i-1} + p_{i-1} c_{i-2}]$$

$$= g_i + p_i g_{i-1} + p_i p_{i-1} c_{i-2}$$

$$c_0 = g_0 + p_0 c_{i-1}$$

$$c_1 = g_1 + p_1 c_0$$

$$= g_1 + p_1 (g_0 + p_0 c_{i-1})$$

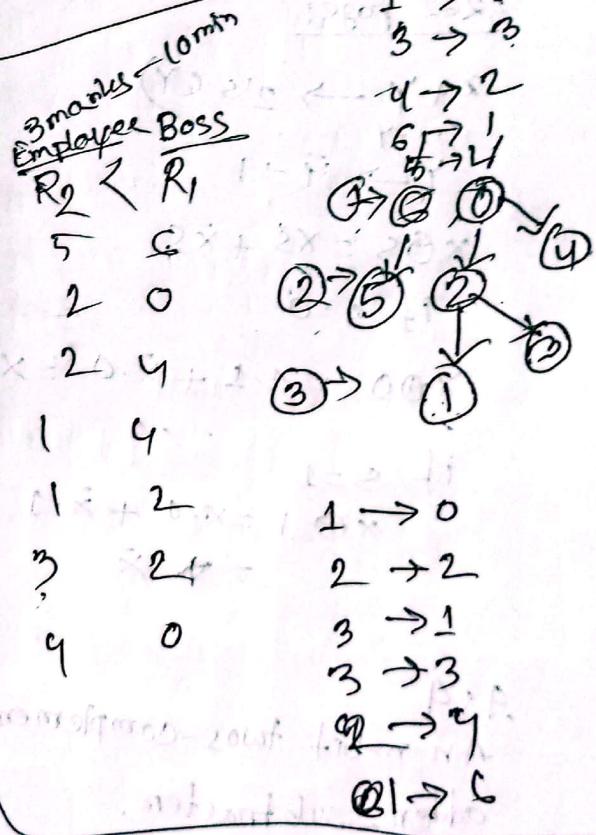
$$= g_1 + p_1 g_0 + p_0 p_1 c_{i-1}$$

fig 9.5 overall structure
Carry-look-ahead adder.

section 9.1: Fixed-point

Arithmetic

• Ripple adder



TOPIC NAME:

DAY:

TIME:

DATE:

10.11.2024

Architecture

- fixed Point Arithmetic
- Low-cost addition and subtraction of two's-complement numbers
 - a) 1 bit adder module
 - b) 8 bit adder-subtractor
- Multiplication

$$P = \sum_{j=0}^3 x_j \cdot 2^j y \quad (\text{manual})$$

$$P_i = P_{i-1} + x_{i-1} \cdot 2^{i-1} y \quad (\text{machine})$$

A_1, A_0

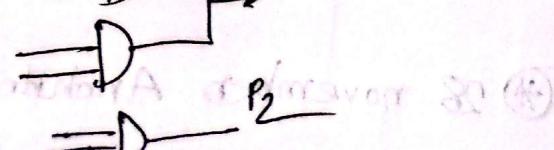
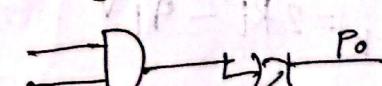
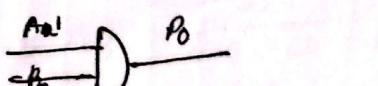
B_1, B_0

$$\begin{array}{r} A_1 \\ A_0 \\ \hline B_1 \\ B_0 \\ \hline A_1 B_0 \\ A_0 B_1 \\ \hline A_1 B_1, (A_1 B_0 + A_0 B_1) \\ A_0 B_0 \end{array}$$

$$P_0 = A_0 B_0$$

$$P_1 = A_1 B_0 + A_0 B_1 + \text{Carry}$$

$$P_2 = A_1 B_1 + \text{Carry}$$



• 1950 → Booth
↳ Booth's Algorithm

• Reducing \oplus

• ARMG → decrease number of addition

$$\begin{array}{r} 1010 \\ 1010 \\ \hline \oplus \rightarrow 2 \end{array}$$

shift → 4

$$\begin{array}{r} 01011 \\ 01110 \\ \hline \oplus \rightarrow 3 \\ \text{shift} \rightarrow 5 \end{array}$$

$$\begin{array}{r} 0011110 \\ \swarrow \quad \searrow \\ \text{start} \quad \text{end}+1 \\ ① \quad ⑦ \end{array}$$

$$\begin{aligned} & 2^{i+1} - 2^j \\ & 2^6 - 2^1 = 64 - 2 = 62 \end{aligned}$$

0110110

$$(2^6 - 2^4) + (2^3 - 2^1) = 98 + 6 = 54$$

TOPIC NAME :

DAY : _____

TIME : _____

DATE : / /

$$\begin{array}{r} 10100 \\ \overline{01011 \quad (M) \quad 0101} \\ 01110 \quad (Q) \\ \hline \end{array}$$

$M \times Q = M$

$$M \times Q = M \times (2^4 - 2^1) \text{ i.e.}$$

$$= 2^4(M) - 2^1(M)$$

$$\begin{array}{r} 0100 \\ 0110 \end{array}$$

$$= 2^4(M) + 2^1(-M)$$

$$\boxed{\text{Left shifting}} = 2^4(M) + 2^1(\bar{M})$$

$$2^4(M)$$

$$= 01011 \times 2^4$$

$$= 010110000$$

$$2^1(\bar{M}) = 10101 \times 2^1$$

$$= 101010$$

$$\downarrow$$

$$(11101010)$$

$$010110000$$

$$111101010$$

$$\hline 101001010$$

Total no. of addition 2

• Division

$$D = Q \times V + R \rightarrow \text{Remainder}$$

- ↳ quotient
- ↳ dividend

$$0 \leq R < V$$

$$\begin{array}{r} 1010 \\ \hline 101 \quad | \quad 10010 \quad | \quad 011 \\ 0000 \quad | \quad 93V \\ \hline 1001 \quad R_1 \\ 101 \quad 932^{-1}V \\ \hline 01000 \quad R_2 \\ 101 \quad 932^{-2}V \\ \hline 11 \quad R_3 \end{array}$$

Right shifting

$$\begin{array}{r} ? \quad D = 2R_0 \\ 101 \quad | \quad 100 \quad 110 \quad | \quad 93 \quad 92 \quad 91 \\ 000 \quad 293V \quad | \quad 0 \quad 1 \\ \hline 100110 \quad R_1 = 2R_0 - 2^0 93V \\ 101 \quad 2R_1 \\ 92V \\ \hline 010010 \quad R_2 = 2R_1 - 92V \end{array}$$

$$R_{i+1} = 2R_i - 9_i V$$

④ 28 November Architecture CT

GOOD LUCK

TOPIC NAME:

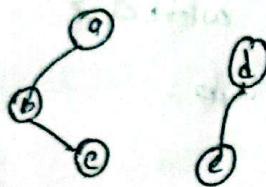
DAY:

TIME:

DATE:

left node connected first
DFS first visit

- DSU (Disjoint set Union)



b need to connect, if not
connect them

12.11.2024
Architecture

4.2

- Arithmetic-logic units.

fig: 4.28

A basic-n-bit arithmetic
logic unit (ALU)

4.29 fig

$$f_i \rightarrow x_i + 4, x_i - 4, x_i \times 1$$

$\hookrightarrow f(x_i, y_i)$

$$m_3 = x_i y_i, m_2 = x_i \bar{y}_i, m_1 = \bar{x}_i y_i$$

$$m_0 = \bar{x}_i \bar{y}_i$$

$$f(x_i, y_i) = s_3 m_3 + s_2 m_2 + s_1 m_1 + s_0 m_0$$

$$f(x_i, y_i) = s_3 x_i y_i + s_2 x_i \bar{y}_i + s_1 \bar{x}_i y_i + s_0 \bar{x}_i \bar{y}_i$$

$$s_3 = 1010 = 0 x_i y_i + \bar{x}_i \bar{y}_i$$

$$s_2 = 0110 = x_i \bar{y}_i + \bar{x}_i y_i$$

Design of a combinational arithmetic
log ic unit

74181 IC

fig 4.30

A register-level view of the

74181 1bit ALU

$$f_i = I_{Pi} \oplus I_{Gi} \oplus (I_{Ci-1} + M)$$

$$I_{Pi} = A_i B_i S_0 + \bar{B}_i S_1$$

$$I_{Gi} = A_i \bar{B}_i S_2 + A_i B_i S_3$$

$$M = 1 :$$

$$f_i = I_{Pi} \oplus I_{Gi} \oplus (I_{Ci-1} + 1)$$

$$= I_{Pi} \oplus I_{Gi} \oplus 1$$

$$\Rightarrow I_{Pi} \oplus \bar{I}_{Gi}$$

$$\begin{array}{l} I_{Gi} \oplus 1 \\ 0 \oplus 1 = 1 \\ 1 \oplus 1 = 0 \end{array}$$

$$M = 0 :$$

$$f_i = I_{Pi} \oplus I_{Gi} \oplus (I_{Ci-1} + 0)$$

$$= I_{Pi} \oplus I_{Gi} \oplus I_{Ci-1}$$

work as an adder

TOPIC NAME : _____

Fig: 9.31 is structure of
basic design (sequential)

- ALU
- basic design Fig. 9.32.

TOPIC NAME:

DAY:

TIME:

DATE:

3 7 9 3 4 7
5 C 3 3 5 6
8 9 7 7 8 9

3 9 7 3 5 C 7 8 9

$$1+1+1+1 = 4$$

19.11.2029
Architecture

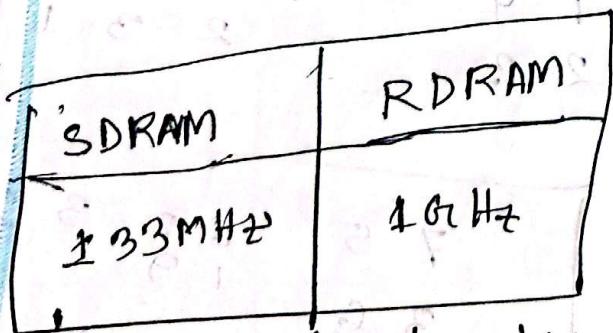
RDRAM

High Speed Memory

To interface Processor
Memory data transfer

Data bus 8-4 pin width

data transfer rate:
- 500 MB/s



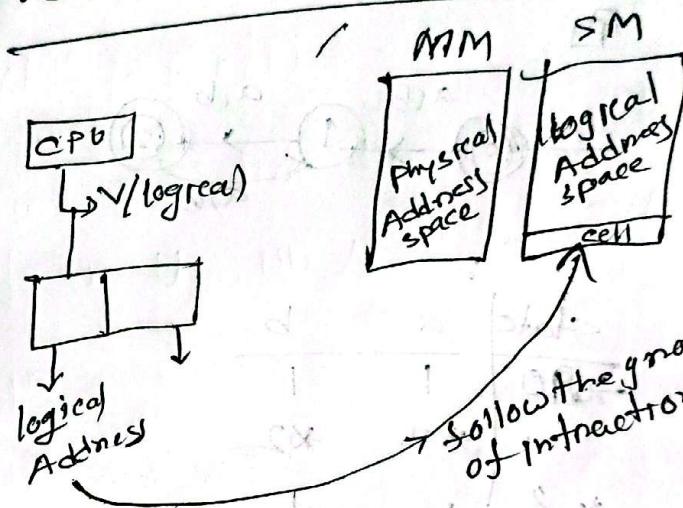
video controller/graphic
card, cache inside CPU,
modem
Server

trig6.14

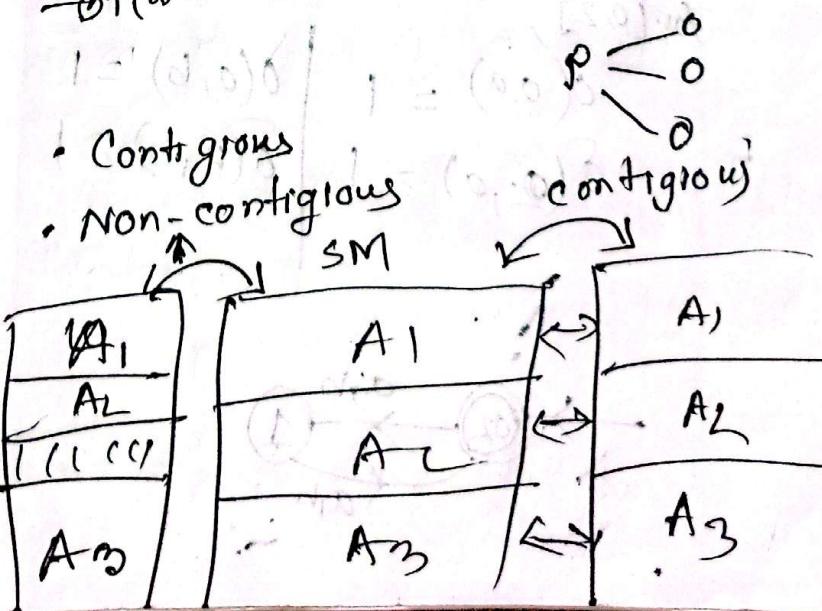
The Rambus DRAM Interface

multilevel memory

Cache and Virtual memory



SM address copy 2²² mm
MM address 2²² address
RA space 2²² address
other address translation



TOPIC NAME:

DAY:

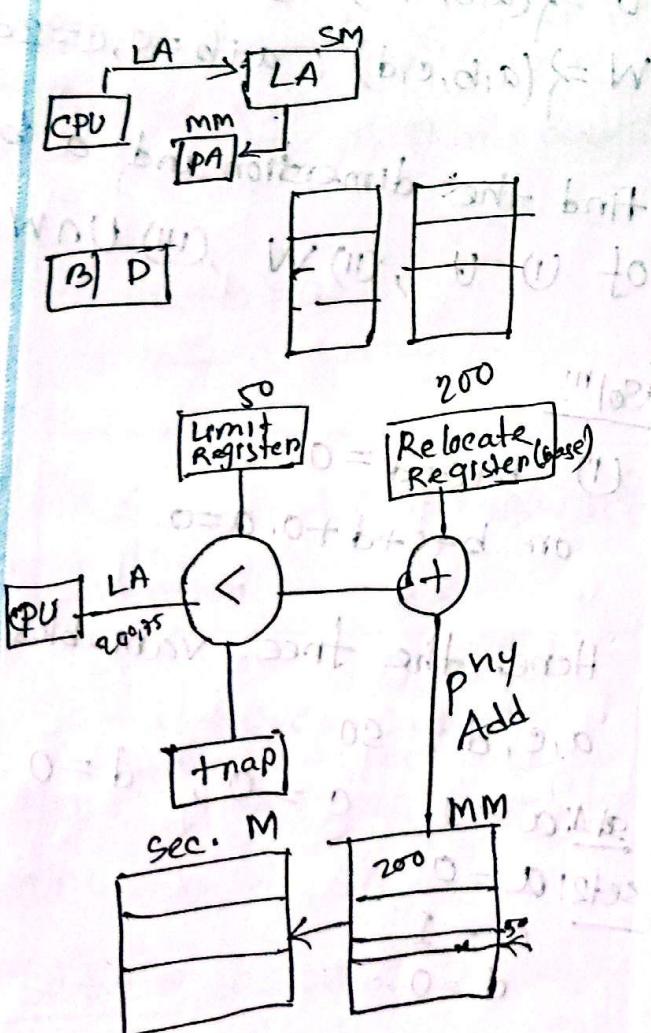
TIME:

DATE: / /

wasted block memory
is called fragmentation

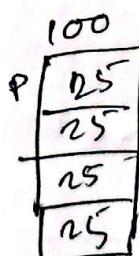
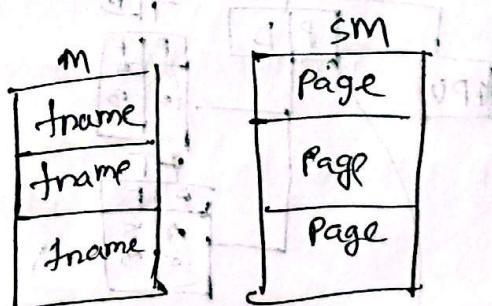
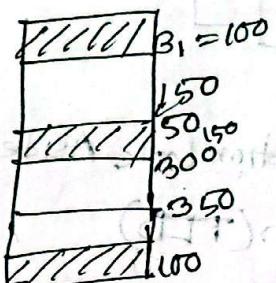
effective address

base address + displacement



Non-contiguous

→ Paging
→ Segmentation

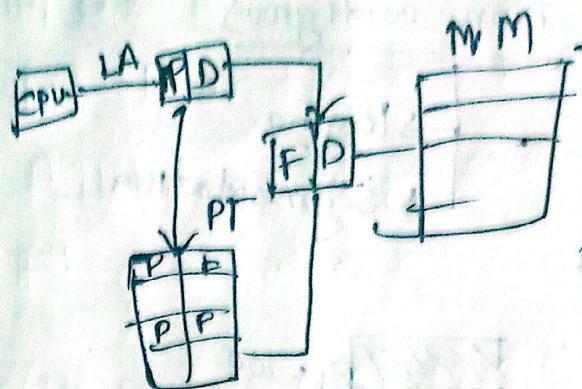


Non-contiguous

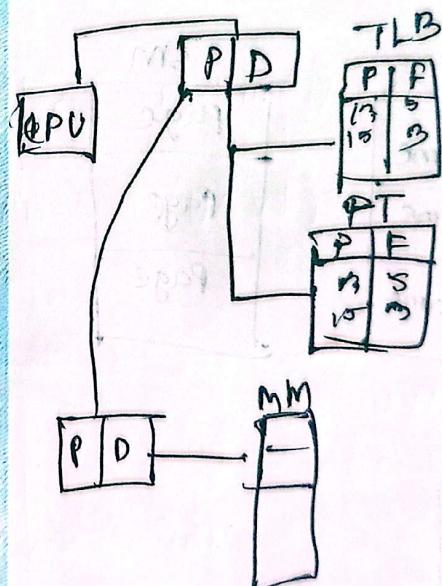
Page table

Page	frame
1	5
2	3

TOPIC NAME :



• Translation Look Aside
Buffer (TLB)



Nonisolate
knowledge gate

TOPIC NAME :

DAY:

TIME:

DATE:

$(0, 0, 1, 0), (0, 0, 0, 1)$

$$-2 - 1 + 3 = 0$$

$$-n_2 - n_1 + n_3 = n_2 = -n_2 - n_1 + n_3$$

$$2 - 3 + 8 = 7$$

$$-1 - 5 + 3 = -3$$

④

$$4 + 3 - 5 = 2$$

$$-2n_2 + n_1 + n_3$$

$$n_3 = 2n_2 + n_1 + n_3 - 4 + 1 + 3$$

$$= 2 \times 3 - 2 + 3$$

$$= -6 - 2 + 8 = 0$$

$$-2 \times 2 + 1 + 3$$

$$-2 \times 1 + 5 - 3 = 0$$

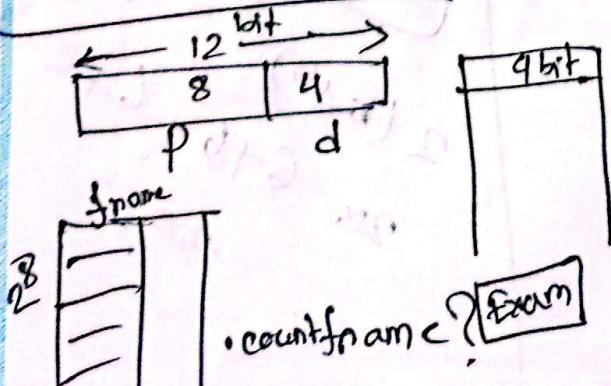
$$-2 \times -4 + 3 - 5$$

$$8 - 8 = 0$$

20.11.2024

Architecture / Entry

Address space $1096 = 2^{12}$



P =

Process size = $9B$

Page size = $2B$

No. of pages: $\frac{9B}{2B} = 2$

main memory

MM			
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Main Memory size = $1GB$

frame size = $2B$

No. of frame: $\frac{1GB}{2B} = 8$

Process:

Page:

0	0	1	P ₀
1	2	3	P ₁

logical Address:

Page no.	displacement
1 bit	1 bit

③ \Rightarrow

1	1
---	---

Physical Address:	
frame no.	displacement
8 bit	1
100	$\frac{1}{4} = 3$
110	$\frac{1}{4} = 5$
P F	
1 4	

- Grate Smashers youtube tutorial
- ⚡ PT = Page table

題 Problem:

Logical Address Space : 4GB
Physical Address Space : 64MB
page size : 4 KB

$$\downarrow \text{No. of. Pages} = 2^{20}$$

No. of frames = 2

No. of Entries in PT^{2^k}

size of PT = $2^{20} \times 14$

$$\text{LAS} = 9 \text{GB} = 2^2 \times 2^{30} = 2^{32}$$

20	4 KB
	$\frac{2^{32}}{2^{12}}$

No. of pages:

$$\text{PAS} = 64\text{MB} = 2^6 \times 2^{20} = 2^{26}$$

No of Entries in $PT = 2^{20}$

④ Max Memory Support = 4GB
 word Addressable, 1W=2byte
 size of Address Bus=?

$$4G_B = A \times 2^B$$

$$2^{32} \text{ byte} = A \times 2^B$$

$$A = 2^{91}$$

070981

(67398)

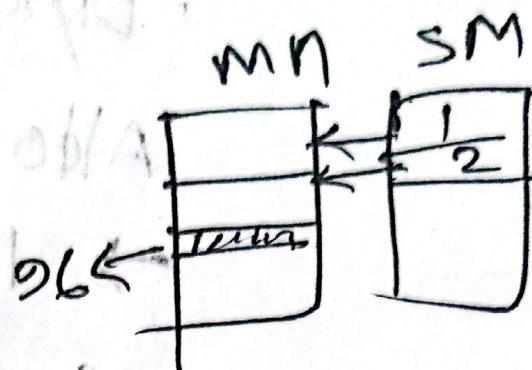
$$\cancel{4+2+2}$$

$$2(a+b) \cdot 6+6+6+c+b$$

26.11.2024

Architecture

Segmentation



total Size : 4096

pagesize : 2048

page : 2

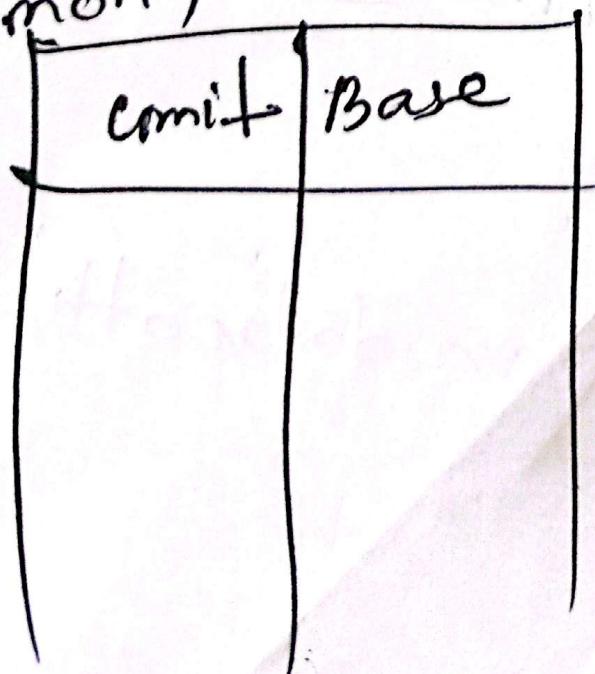
$$\frac{4096}{2048} \rightarrow 4 \text{ pages}$$

96

fragmentation

mangement :

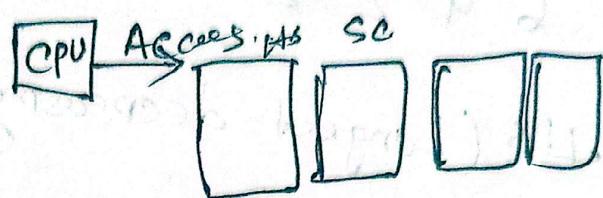
Memory



TOPIC NAME:

• Hardware Segmentation

- Memory Allocation (44B) page
- Hashing



• demand swapping

• Replacement policy

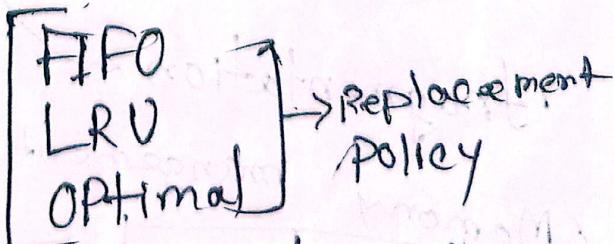
• Dynamic Memory Allocation

→ Nonpreemptive

→ preemptive

• first fit

• Best-fit



ALU
processor datapath

$$\log_2^2 = 1$$

TOPIC NAME:

DAY:

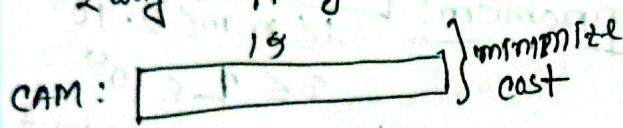
TIME:

DATE:

- Organization of CAM word in Fully Associative Mapping

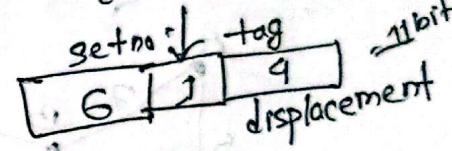
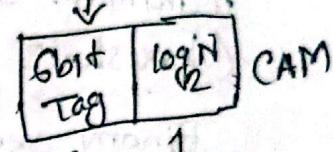
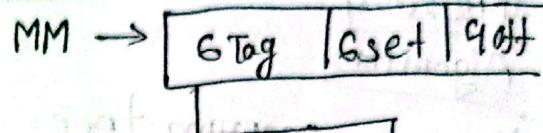
(3) Set-Associative:

2 way mapping



2 way Mern. $\frac{128}{2} = 64$ sets (6bit)

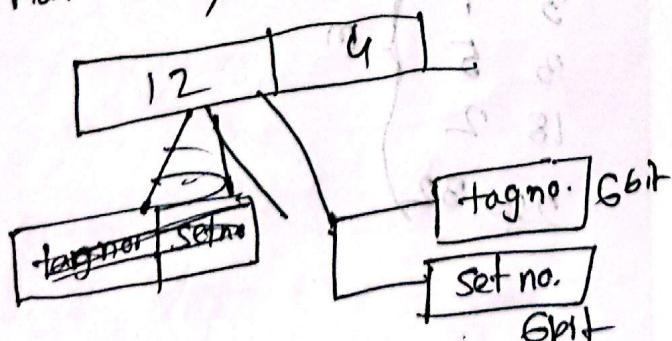
$MM \rightarrow \frac{4096}{64} = 64$ slots [column].
Tag no.



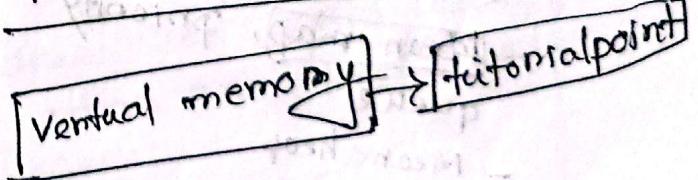
$$\log_2^4 = 2$$

set no. tag displacement
 $5 | 2 | 4 = 11$ bit

Main Memory Address:



H.W form 4 way (tutorial point)



Find 4033

$$64) 4033 (63 \text{ (Tag no.)} \\ 4032 \quad (11111)_2 \\ \underline{-1} \\ \text{set no. } (000001)_2$$

TOPIC NAME:

DAY:

TIME:

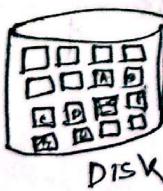
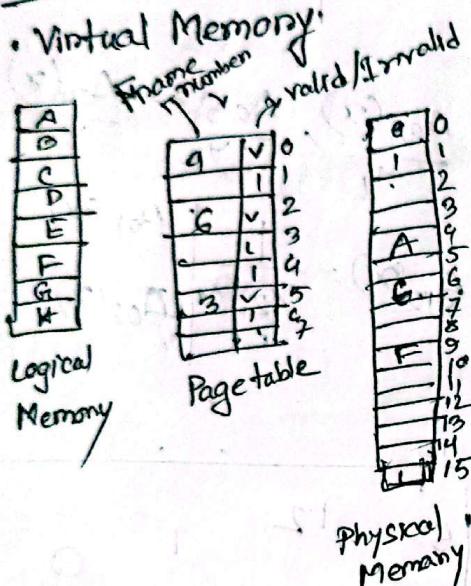
DATE:

03.12.2024
Algorithm

03.12.2024

Architecture

• Virtual Memory



- Page will be brought into memory only when it is needed.
 - Less I/O needed

$A = A + B$ (control signal design)

Virtual memory (or page fault 20th)

① Page fault

• steps in handling a page fault

Performance of Demand Paging

* $0 \leq \text{Page fault probability } (P) \leq 1$

* Effective access time (EAT)

$EAT = (1-P) \times \text{memory access} + P \times \text{page fault overhead}$

• Swap

Chapter 5

② Control Design

- Basic Concepts

fig 5.1 Processor composed of a datapath unit DP and a control unit CU.

$$A = A + B$$

(control signal design)

③ { HW control
Macroprogram named

GOOD LUCK™

TOPIC NAME:

DAY:

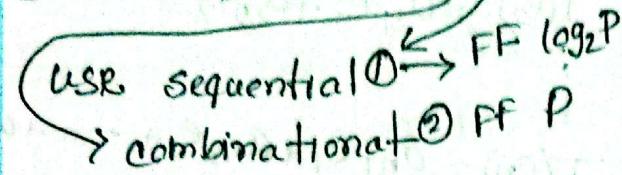
TIME:

DATE:

② types method

① Classical Method

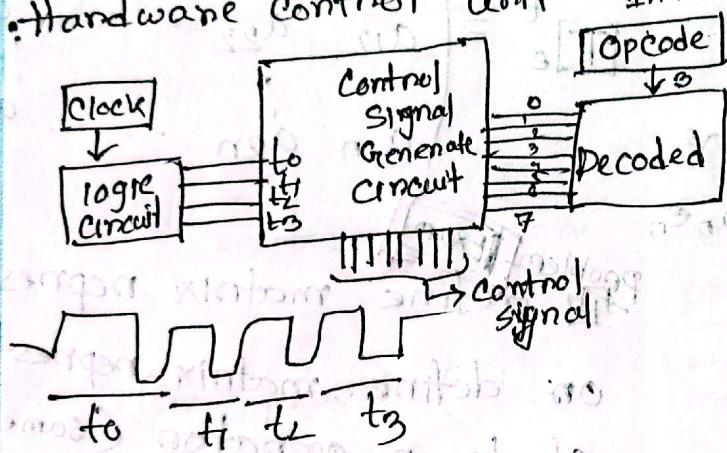
② One off method



P = number of stage

- ① complex and less hubbend of FF.
- ② simple and easy for finding.

Hardware control unit



5.1.2 Hand wane control

state tables

fig 5.9: state tables for a finite-state machine

(a) Mealy type, (b) Moore type

(a) depends on input and state.

(b) depends on state

③ GCD processor:

fig 5.5:

Procedure gcd to compute the greatest common divisor of two numbers.

fig 5.6 circuit

fig 5.7 state table

next class: Excitation table
logical expression
and Method

- Classical Method:

fig 5.8: Excitation table for the control unit of the gcd processor.

1001010

$$D_1^+ = (\overline{X_R} \cdot D_0) + (X_R \geq Y_R) + D_0 \quad \textcircled{1} \quad \textcircled{1} \quad \textcircled{1} \quad \textcircled{1}$$

$$\text{subtract} = D_1 \cdot \bar{D}_0$$

$$\text{swap} = D_1, D_0$$

fig: 5.9: All-NAND classical design
for the control unit of the gcd processor.

H.W
One-hot method (One ff per state)
Assignment

- Example codes for four states:

$$(Y_3, Y_2, Y_1, Y_0) = 0001, 0010, 0100, \text{ and } 1000$$

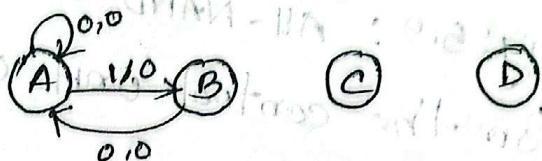
- Only include variable that is 1 in equations.

- State with code 0001, is represented by equations by Y_0 instead of $\bar{Y}_3 \bar{Y}_2 \bar{Y}_1 Y_0$ since codes with a 0 on over two 1s

have don't care next
state values.

- Provides simplified analysis and design
- Combinational logic may be simpler, but flip-flops cost more
- Thus, design may or may not be lower cost.

④ State assignment:



④ Truth table to kmap

④ One hot Assignment: Circuit

→ process on classical

Method

One hot method

Assignment

TOPIC NAME:

DAY:

TIME:

DATE: / /

08.12.2024

Architecture

gcd alu

Melly-Mone

Chapter 7
system

- computer system
- system component
- processor system

- communication methods
- intensystem, intrasystem
- Beises

fig 7.1
communication within a computer via a single shared bus

powerPC 603 micro-processor.

fig 7.3
computer with separate system and IO buses



fig 7.9

The small-computer SCI IO bus

5M B/s.

• Bus interfacing (50% page)
long

• tri state buffer

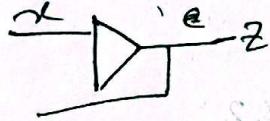


fig 7.13

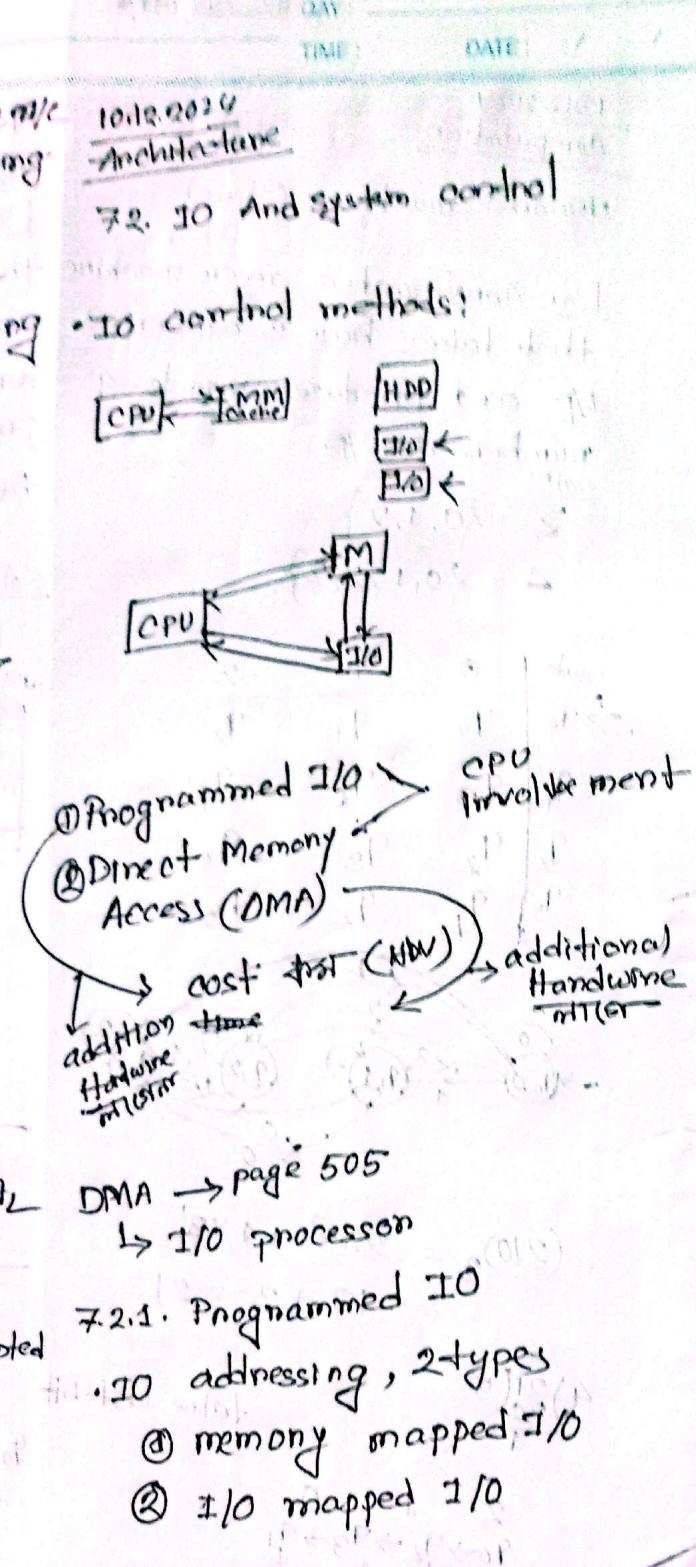
fig 7.14:
communication with circuit 1 conduct

2. Bus arbitration

• Bus step

- ① Daisy chain (fig. 7.20)
- ② Polling
- ③ Independent Request

fig 7.20: bus arbitration using daisy chaining.



TOPIC NAME

1. Motorola, 2. Intel
 2. Intel

fig. 7.26:

fig 7.31: The 8255 programmable peripheral interface circuit.

7.2.2. DMA and Interrupts

fig 7.34

DMA and interrupt breakpoint during instruction processing

fig 7.35

circuity required for direct memory access (DMA)

• Interrupts

fig 7.36 single line interrupt
 fig 7.37 multi line

④ Requests + Interrupts

NB

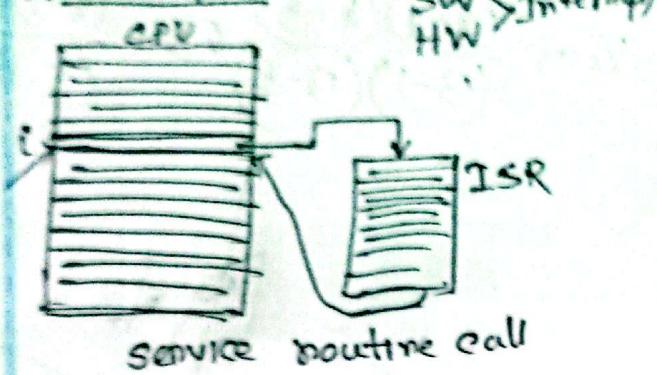


TOPIC NAME

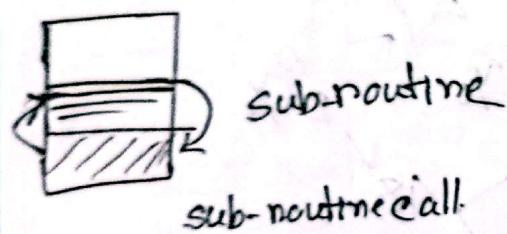
11-12-2024

Architecture

④ Interrupts



ISR = Interrupt Service Routine



⑤ Interrupt service routine
किसी वार्ता के द्वारा:

2 Methods to find the location of ISR.

→ Vectored Interrupts

→ Non-vectored Interrupts

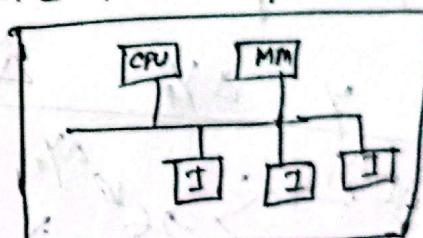
Non-vectored:

- Address of ISR is stored in a specific memory location.

• Device does not send any address

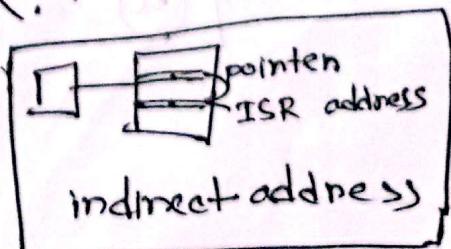
Vectored:

- The source of interrupt supplies an interrupt vector along with the interrupt.



(1) Interrupt vector gives the first address of ISR direct

(2) Interrupt vector gives a pointer to the address of ISR.



IO processor, HDL, control unit multiplex

DAY:

TIME:

DATE: / /

15.12.2024

Architecture4.1.3DivisionDivisor: V divided: D Quotient: Q Remainder: R

$$D = Q * V + R \quad [0 \leq R < V]$$

$$\frac{D}{V} = Q + \frac{R}{V}$$

$$\frac{17}{3} = 5 + \frac{2}{3}$$

$$\begin{array}{r}
 \textcircled{1} \quad 100110 \\
 \textcircled{2} \quad 101 \quad | \quad 100110 \quad | \quad \textcircled{3} \\
 \quad 000 \\
 \hline
 \quad 100110 \\
 \end{array}$$

$q_3 V^2$
 R_1
 $q_2 V^2$
 R_2
 $q_1 V^2$
 R_3
 $q_0 V^2$
 R_4

$$R_{i+1} = R_i - q_i 2^{-l} V$$

$$R_{i+1} = 2R_i - q_i V$$

$$\begin{array}{r}
 101 \left| \begin{array}{r} 100110 \\ 000q_2V \\ \hline 100110 \end{array} \right| 01 \\
 100110 \quad R_1 \\
 1001100 \quad 2R_1 \\
 101 \quad q_2V \\
 \hline 100100 \quad R_2 \\
 1001000 \quad 2R_2 \\
 101 \quad q_2V \\
 \hline 100000 \quad R_3 \\
 1000000 \quad 2R_3 \\
 101 \\
 \hline 011000 \quad R_4
 \end{array}$$

fig 4.23

the datapath of a sequential n-bit binary divider.

fig 4.26

A cell D for array implementation of nesting division

t = borrow in

u = borrow out

z = output controlled a

when $a = 1$

$$z = x \oplus a \cdot (y \oplus t)$$

$$= x \oplus 0 \cdot (y \oplus t)$$

$$= x$$

when $a = 0$

$$z = x \oplus a \cdot (y \oplus t)$$

$$= x \oplus 0 \cdot (y \oplus t)$$

$$= x \oplus y \oplus t \text{ (Subtraction)}$$

carryout

$$u = z \cdot y + x \cdot t + y \cdot t$$

$$u_i = 0,$$

$$\text{output} = 2R_i$$

$$\text{and } q_i = \overline{u_i} = 1$$

$$u_i = 1, \text{ output } 2R_i$$

$$q_i = \overline{u_i} = 0$$

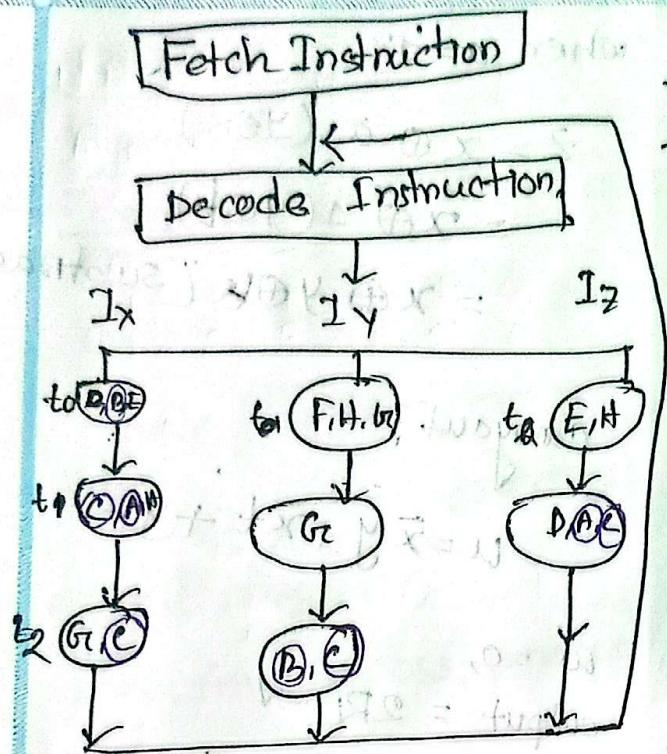
Any prime number control unit design

Microprogrammed Control unit design

flowchart

TOPIC NAME : _____

15
Al



$$A = I_x t_1 + I_z t_1 \\ = t_1 (I_x + I_z)$$

$$B = I_x t_0 + I_y t_2$$

$$C = (I_x + I_z) t_1 + (I_x + I_y) t_2$$

tutorial point video
decker lagbe

GOOD LUCK™

TOPIC NAME : _____

17.12.2024

Architecture

FEC

Easy Engineering Class,

• part 1 Question

• part 2

• part 3

mapping (2020 Exam 4)

math atleast must

reso accadacy

memory mapping math

• Direct, Memory Mapping

solved Examples

Shanu kuttan CSE

classes

L-3, 13

Cache Mapping

Questions | Memory

Organisation

TOPIC NAME:

DAY:

TIME:

DATE: / /

18.12.2024 Architecture
② shanu

Solved problem in control unit
in Computer architecture

③ Engineering Funda [20]

① Easy Engineering classes

3. 1, 2, 3, 5, 6, 18, 19

7X, 8X, 9X, 10X, 11X

12X, 13X, 14X, 15X 16X

17X

Parameters of pipelining in

Computer Organization &

Architecture

Exam

memory - 2, mapping (math
Allocation, cache mapping,
virtual, CPU, performance,
control unit, micro-unit,
hardware control unit design,
one-hot, classical method
system organisation, interrupts
arithmetic logic unit design

$$T = \frac{1}{f}$$

TOPIC NAME :

DAY:

TIME:

DATE:

DMA,
Program I/O
Interrupts

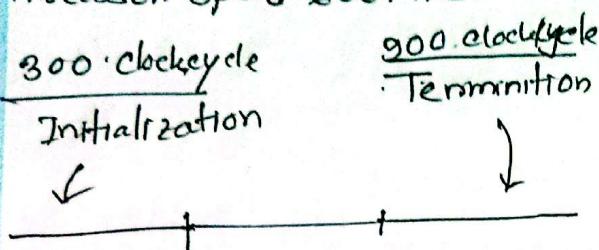
$$= 2000 \mu\text{s}$$

$$\% \text{ of consume time} = \frac{2}{2000} \times 100$$

$$= 0.1\%$$

Problem:

HDD Transfer rate 10MB/sec
Processor speed 600MHz.



If the size of ^{HDD} Transfer
size = 20KB

% of consume time = ?

Soln:

$$F = 600 \times 10^6 \text{ Hz}$$

$$\frac{T}{\text{cycletime}} = \frac{1}{F} = \frac{1}{600 \times 10^6} \text{ sec}$$

DMA (Init + Ter) time = ~~600~~
= 900

$$= (900 + 300) \times \frac{1}{600 \times 10^6}$$

$$= 2 \mu\text{s}$$

HDD transfer rate 10MB/sec
1 Byte " time = $\frac{1}{10 \times 10^6}$

$$= \frac{1}{10^7} \text{ sec}$$

$$20 \text{ KB} " " = \frac{20 \times 10^3}{10^7} \text{ sec}$$

Problem:

Device $\xleftrightarrow{\text{connected}}$ CPU

Device data transfer rate = 10KB/sec

$$\frac{\text{So many transfers}}{\text{Performance gain}} = \frac{\text{Programmed I/O time}}{\text{Interrupt time}}$$

10KB transfer time = 1 sec

$$1 \text{ Byte} = \frac{1}{10 \times 10^3} = 100 \mu\text{s}$$

Programmed I/O time \Leftrightarrow

= time to check status + transfer time

$$= 100 \mu\text{s} + 0 \quad [\because \text{data transfer time negligible}]$$

$$= 100 \mu\text{s}$$

Interrupts I/O time

= Interrupt overhead time + transfer time

$$= 4 \mu\text{s} + 0$$

$$= 4 \mu\text{s}$$

TOPIC NAME :

DAY

TIME:

DATE: / /

$$\text{Performance gain} = \frac{\text{Programmed time}}{\text{Interrupt time}}$$

$$= \frac{100}{4}$$

$$= 25$$

Problem :Consider

device speed = 2 MBPS

2 usec transfer 1 G byte Data
to memoryDMA \rightarrow time CPU Block.

Sol:

$$2 \text{ MB data prepared in 1 sec}$$

$$1 \text{ G byte } " " \frac{1}{2 \text{ MB}} \times 16 \text{ byte}$$

$$= 8 \text{ usec}$$

$$\text{DMA time} = \frac{2 \mu\text{sec}}{8 \mu\text{sec}} \times 100$$

$$= 25 \%$$

* state diagram use for
one hot Diagram (int)
circuit

$$0 + 22 \mu\text{sec} =$$

$$22 \mu\text{sec} =$$

Lagrange

Vector divided