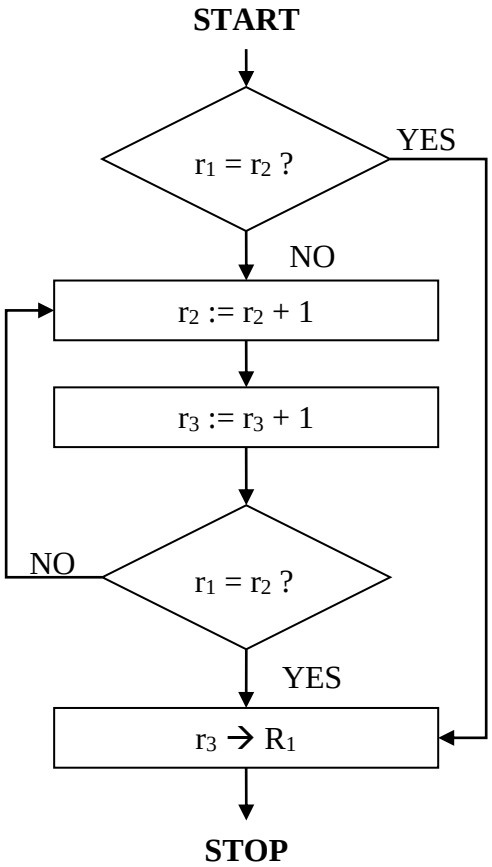
Lecture 1:Overview

Q1 Consider the following program:

I_1: J(1, 2, 6)
I_2: S(2)
I_3: S(3)
I_4: J(1, 2, 6)
I_5: J(1, 1, 2)
I_6: T(3, 1)

i) Analyze the function of the program and Draw the flow diagram for the program
ii) Perform computation under this program with the initial configuration 8,4,2,0, …

Assume the initial configuration is : x,y,z. After k iterations, the value of first register is z+k.
Output is in first register. If x = y + k. then the output is z + x - y



Typical Configuration

x	y	z	

After k cycles round the loop

x	y+k	z+k	

If x = y + k :

z+k	y+k	z+k	

	R1	R2	R3	R4	R5		Next Instruction
Initial configuration	8	4	2	0	0	…	I_1
	8	4	2	0	0	…	I_2 (since $r_1 \neq r_2$)
	8	5	2	0	0	…	I_3
	8	5	3	0	0	…	I_4
	8	5	3	0	0	…	I_5 (since $r_1 \neq r_2$)
	8	5	3	0	0	…	I_2 (since $r_1 = r_1$)
	8	6	3	0	0	…	I_3
	8	6	4	0	0	…	I_4
	8	6	4	0	0	…	I_5 (since $r_1 \neq r_2$)
	8	6	4	0	0	…	I_2 (since $r_1 = r_1$)
	8	7	4	0	0	…	I_3
	8	7	5	0	0	…	I_4
	8	7	5	0	0	…	I_5 (since $r_1 \neq r_2$)
	8	7	5	0	0	…	I_2 (since $r_1 = r_1$)
	8	8	5	0	0	…	I_3
	8	8	6	0	0	…	I_4
	8	8	6	0	0	…	I_6 (since $r_1 = r_2$)
Final configuration	**6**	8	6	0	0	…	I_7 : STOP

Q1.5: Computer Organization vs Architecture

…. is related to Organization or Architecture?

Organization **differs** between different versions

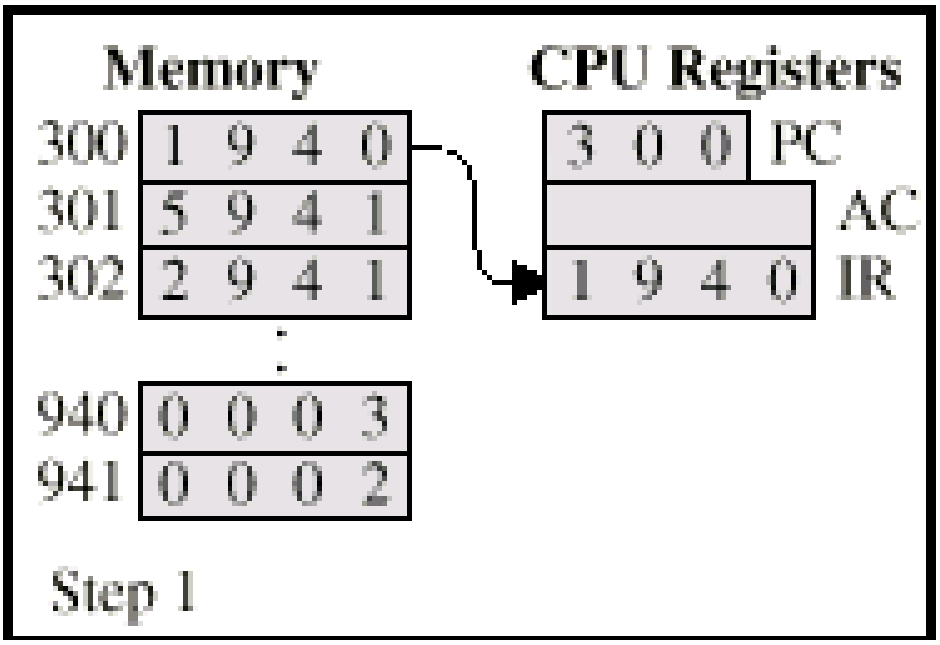
Lecture 4: Assembly level organization= Von Neumann

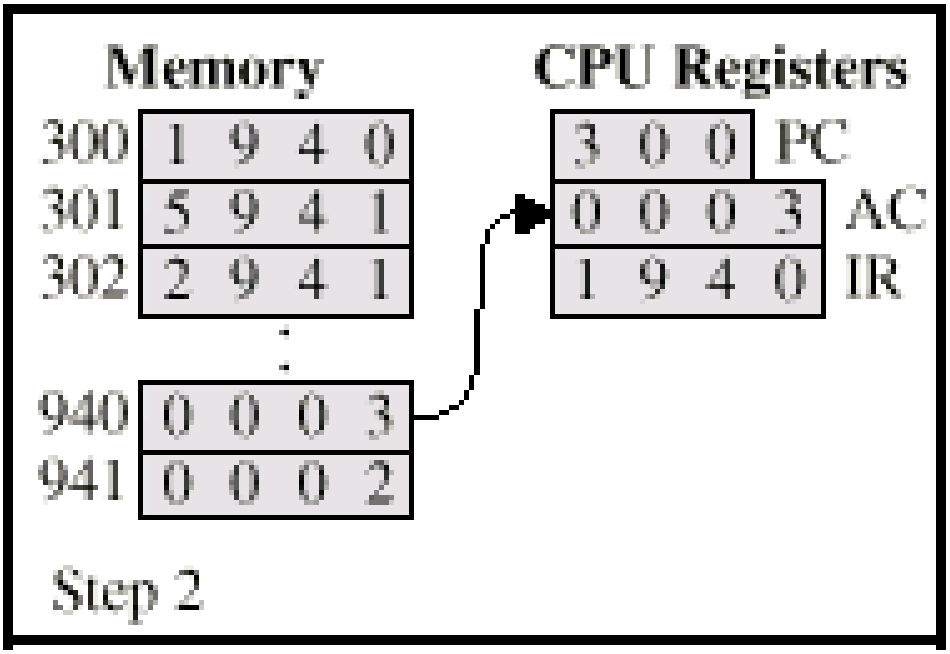
Q2 Trace the Program Execution (**Op-Codes**: 1=load, 2=store, 5=add, …)

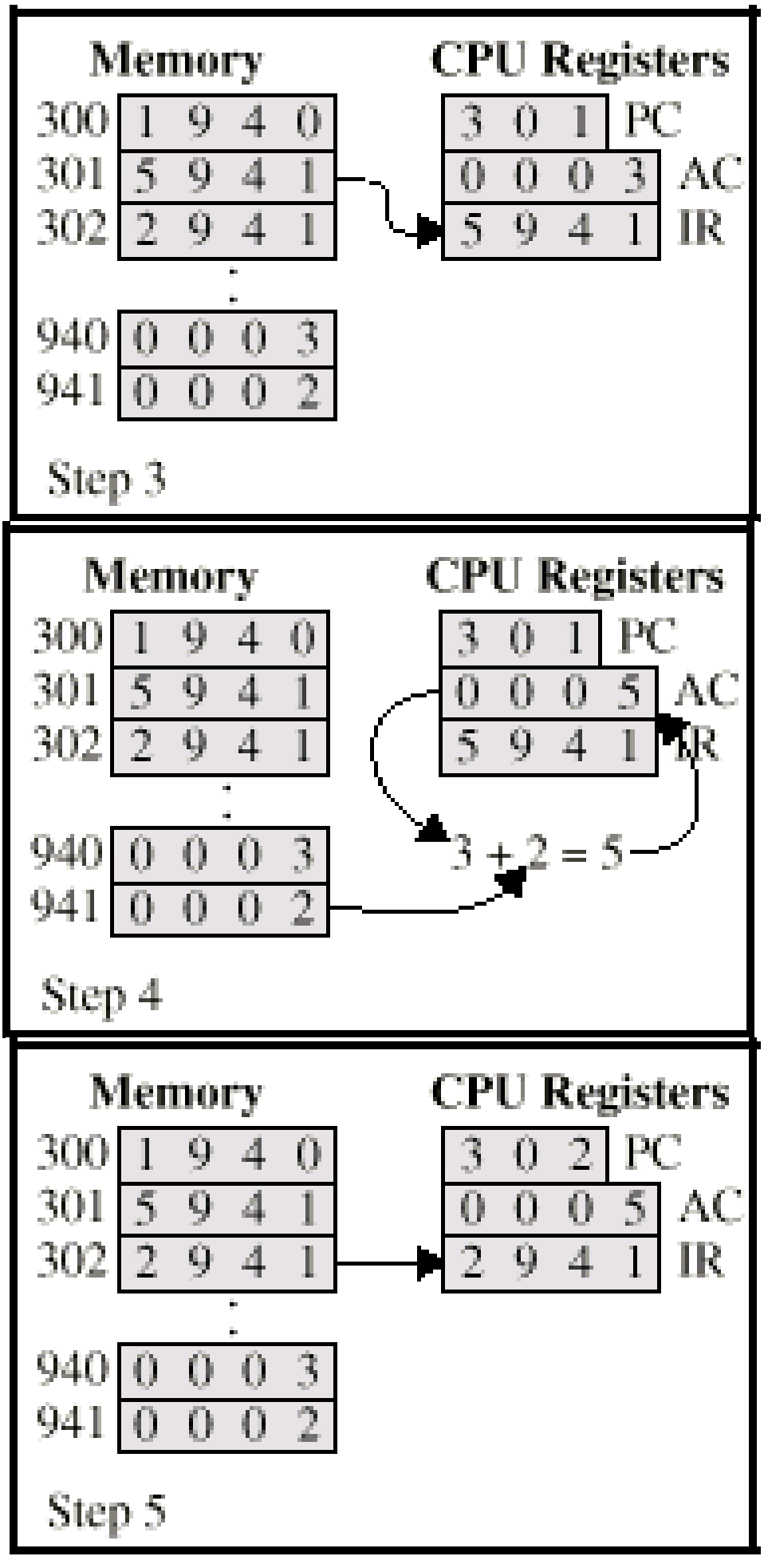
	Memory
300	1 940
301	5 941
302	2 941
	:
940	0003
941	0002
942	0001

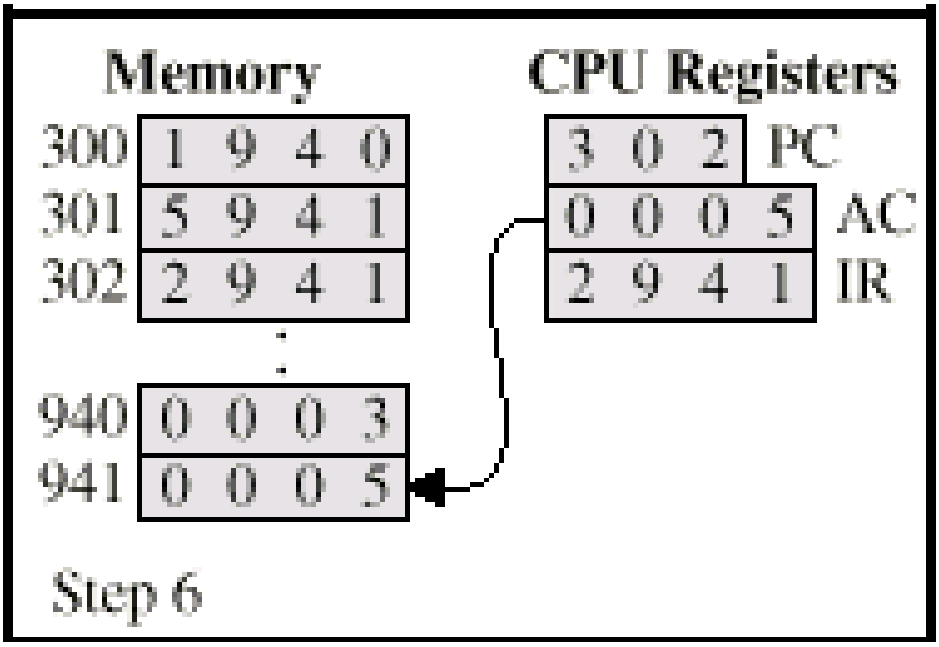
CPU Registers	
	PC
	IR
	AC
	MAR
	MBR

Answer :






Memory
CPU Registers
300 1 9 4 0
301 5 9 4 1
302 2 9 4 1
940 0 0 0 3
941 0 0 0 2
3 0 1 PC
0 0 0 3 AC
5 9 4 1 IR
Step 3
Memory
CPU Registers
300 1 9 4 0
301 5 9 4 1
302 2 9 4 1
940 0 0 0 3
941 0 0 0 2
3 0 1 PC
0 0 0 5 AC
5 9 4 1 IR
3 + 2 = 5
Step 4
Memory
CPU Registers
300 1 9 4 0
301 5 9 4 1
302 2 9 4 1
940 0 0 0 3
941 0 0 0 2
3 0 2 PC
0 0 0 5 AC
2 9 4 1 IR
Step 5

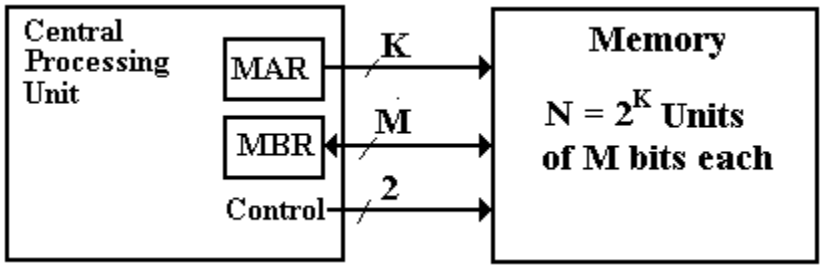


Lecture 6+7:Memory systems

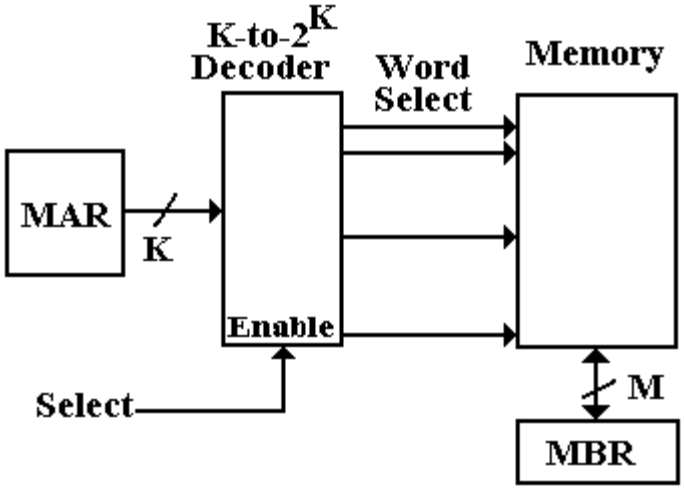
Memory access time → MAR. **Memory cycle time** → MBR.

Q3Design 4GB byte-addressable memory showing length of MAR (K) & MBR (M)?
Or Design 1MB byte-addressable memory showing the length of K and M?

Monolithic view of the memory is shown in the following figure.



The **linear view**

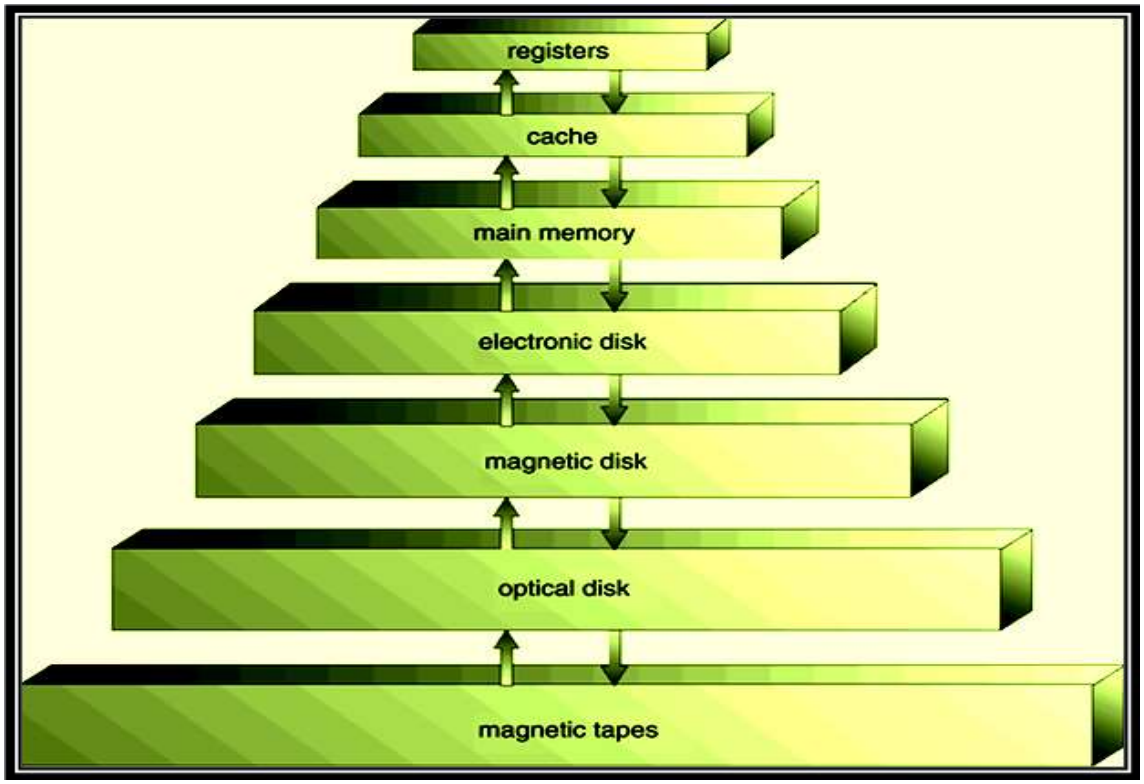


M= 8 bits

- A 1MB memory would use a 20–to–1,048,576 decoder, as 2^{20} = 1,048,576. ➔ K=20 bits
- A 4GB memory would use a 32–to–4,294,967,296 decoder, as 2^{32} ➔ K=32 bits

KB= 2^{10}, MB= 2^{20}, GB=2^{30},
4GB= $4\times2^{30} = 2^2\times2^{30} = 2^{32}$

Q3.6 the higher level in the storage Hierarchy has higher speed (True)



Storage systems organized in hierarchy: Speed, Cost, Volatility

Q3.7: Consider the following page table:

Page	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?

a. 0,430
b. 1,10
c. 2,500
d. 3,400
e. 4,112

Answer:

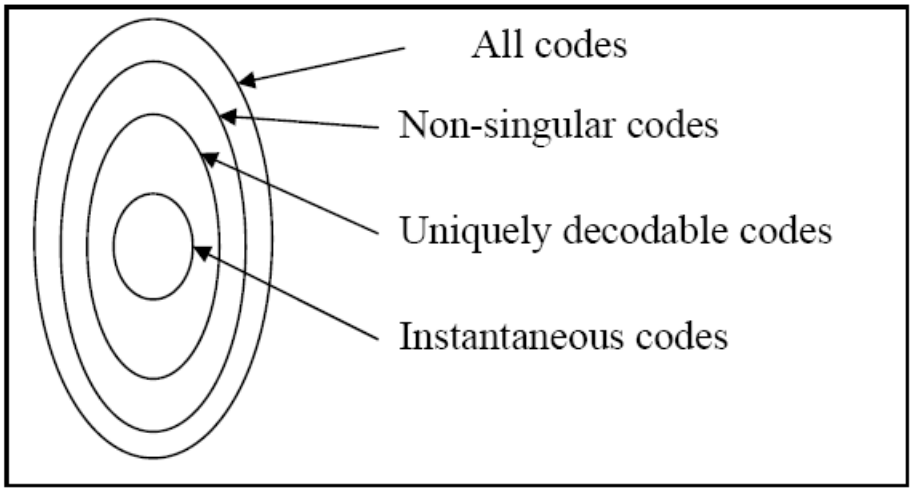
a. 219 + 430 = 649
b. 2300 + 10 = 2310
c. illegal reference, trap to operating system
d. 1327 + 400 = 1727
e. illegal reference, trap to operating system

جزء ملغي

Information theory answers two fundamental questions in communication theory:

- what is the ultimate data compression (answer is the *entropy H*), and
- what is the ultimate transmission rate of communication (answer is the *channel capacity C*) .

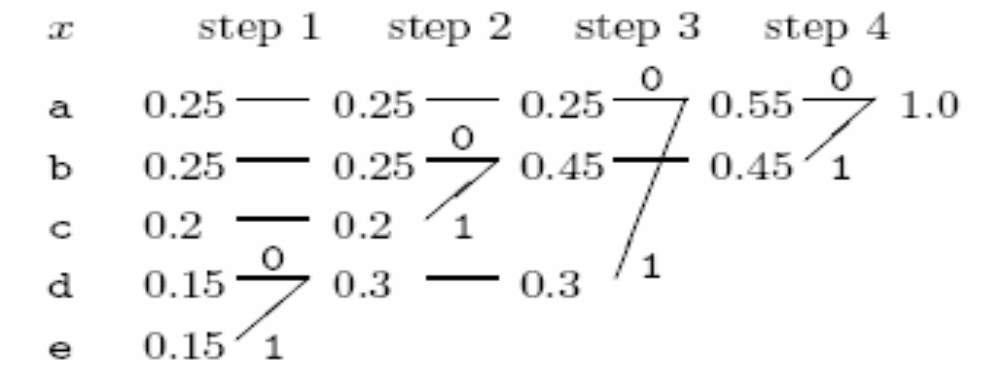
Classes of Codes



Text= aaaaabbbbbccccdddeee

(Huffman coding)

- X ={ a, b, c, d, e }
- P(X) ={ 0.25, 0.25, 0.2, 0.15, 0.15 }.



- The final codes are {00, 10, 11, 010, 011}

a_i	p_i	$h(p_i)$	l_i	$c(a_i)$
a	0.25	2.0	2	00
b	0.25	2.0	2	10
c	0.2	2.3	2	11
d	0.15	2.7	3	010
e	0.15	2.7	3	011

Q3.8: physical and logical

Divide **physical** memory into fixed-sized blocks called **frames** (size is power of 2).

Divide **logical** memory into blocks of same size called **pages**.

Consider a **logical** address space of **8 pages** of **1024 words** each, mapped onto a **physical** memory of **32 frames**.

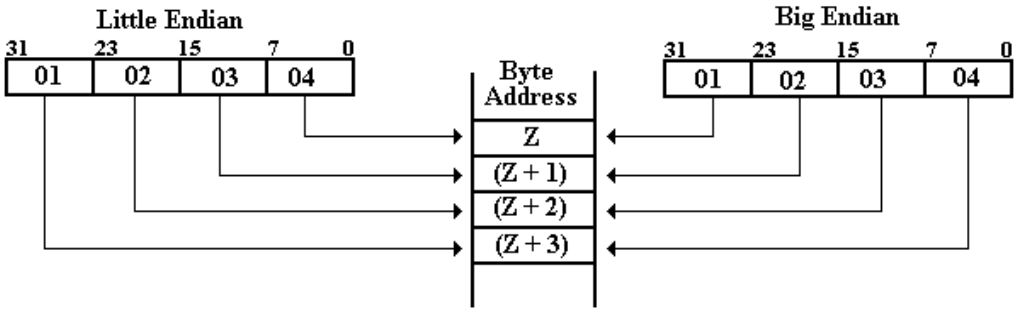
a. How many bits are there in the **logical** address?
b. How many bits are there in the **physical** address?

Answer
Addressing within a 1024-word page requires 10 bits because $1024 = 2^{10}$. Since the logical address space consists of $8 = 2^3$ pages, the logical addresses must be 10+3 = 13 bits. Since there are $32 = 2^5$ physical pages, phyiscal addresses are 5 + 10 = 15 bits.

---

1. ____2
2. ____4
3. ____8
4. ____16
5. ____32
6. ____64
7. ____128
8. ____256
9. ____512
10. ____1024

Q3.9: Big-Endian and Little-Endian



There seems to be no advantage of one system over the other. Big–endian seems more natural to most people. Big-endian computers include the IBM 360 series, Motorola 68xxx, and SPARC by Sun.

Little-endian computers include the Intel Pentium and related computers.

The big-endian vs. little-endian debate shows in file structures when computer data are "serialized" –

Little-endian	Windows BMP, MS Paintbrush, MS RTF, GIF
Big-endian	Adobe Photoshop, JPEG, MacPaint

Lecture 9: Advanced Memory systems

Just Add Memory (٠)
memcomputing (١)
the central processing unit (٢)
hard drive (٣)
(٤) المقاومة الذاكرية هي مقاومة تعتمد قيمتها على مقدار التيار الذي مرّ عبرها سابقا.
(٥) المكثفة الذاكرية هي مكثفة تعتمد سعتها الحالية على قيمة الشحنة التي خزنت فيها سابقا.
(٦) الملف الذاكري هو ملف يعتمد تحريضه الحالي على قيمة التيار الذي مرّ عبره سابقا.
memcomputer (٧)

ويحصل هذا النقل ذو الخطوتين بالاتجاهين لأن ذاكرة الحاسوب لا تستطيع حاليا معالجة البيانات، ولا تستطيع وحدات المعالجة حفظ البيانات. وهذا توزيع شائع للعمل، وهو يحصل في أفضل الحواسيب التي تقوم بأسرع أنواع

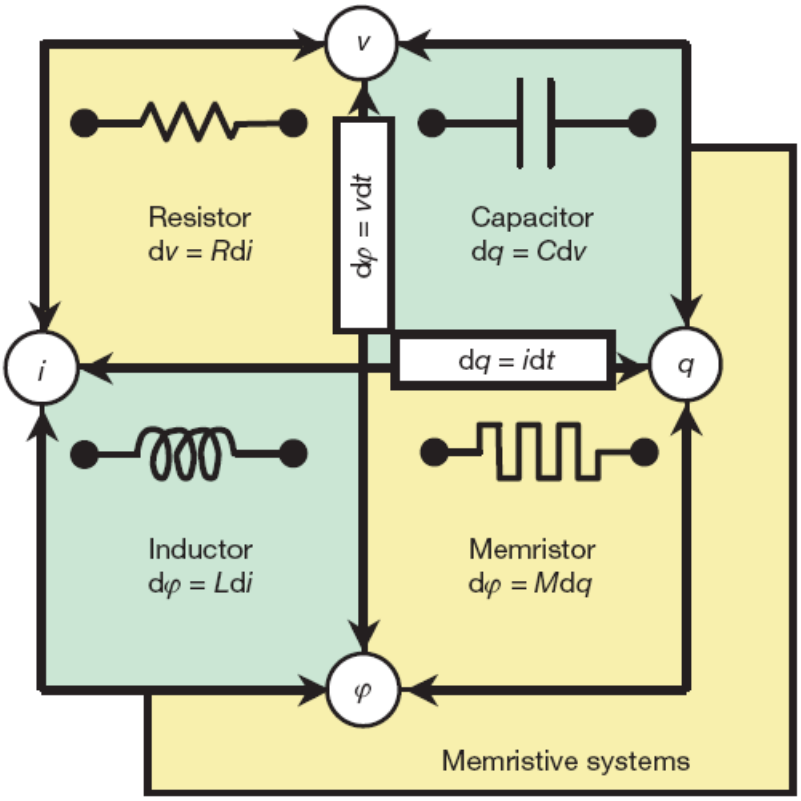
باختصار

في جميع الحواسيب الحديثة تستعمل وحدة للعمليات الحسابية، ووحدة ذاكرة منفصلة توضع فيها البرامج والبيانات. وتستهلك حركة البيانات الكوكبية ذهابا وإيابا بينهما كثيرا من الوقت والطاقة.

لكن تصميما جديدا، يسمى **حوسبة ذاكرية**[1]، يعمل بطريقة تشبه طريقة عمل العصبونات في الدماغ البشري، حيث توجد كلتا وحدتي الحساب والخزن ضمن الوحدة الفيزيائية نفسها.

ويمكن لهذا أن يعني قفزة كبرى في سرعة الحاسوب وكفاءته، إضافة إلى بنيان حوسبة جديد. لذا يحاول العلماء معرفة أفضل الطرق لاستعمال مكونات الحوسبة الذاكرية المختلفة.

Q4: what are the **four** fundamental two-terminal circuit elements?
Resistor, capacitor, inductor and memristor.



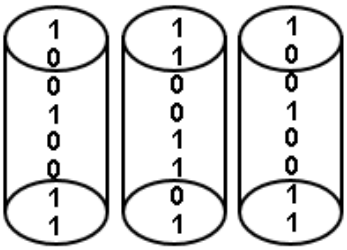
Lecture 10: Interfacing and communication

Q4.1: .. is synchronous or asynchronous?

Characteristics of I/O Devices

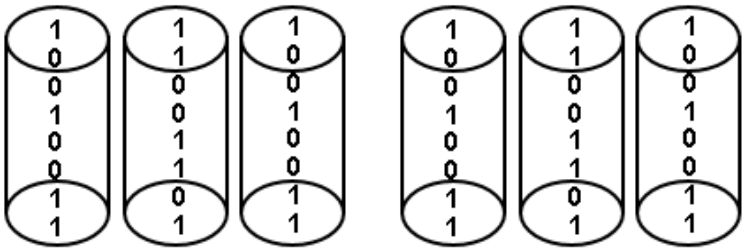
aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only readÐwrite	CD-ROM graphics controller disk

Q4.2: Apply RAID1 and RAID3 on the following data

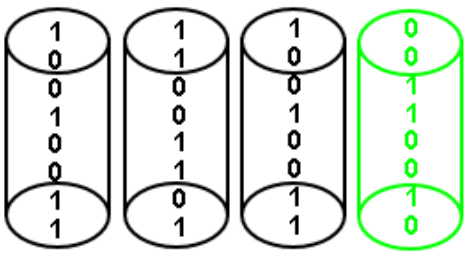


Answer

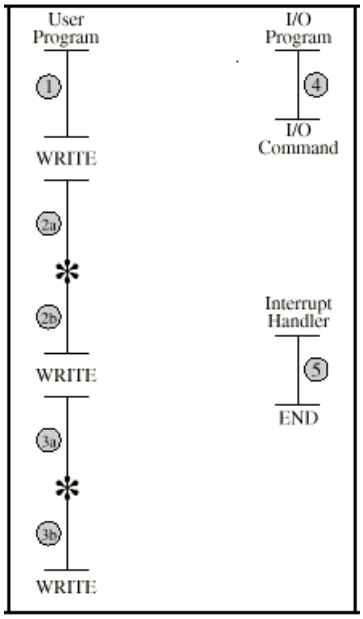
RAID1



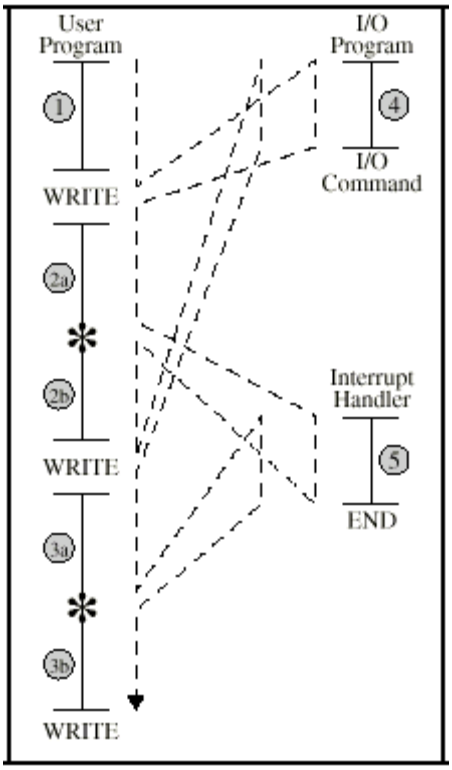
RAID3



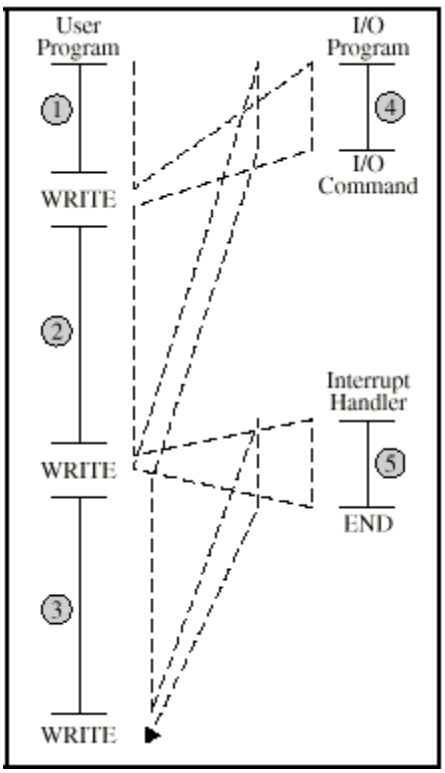
Q4.2: compare between short IO wait and long IO wait, by drawing a line that shows the execution path



Answer



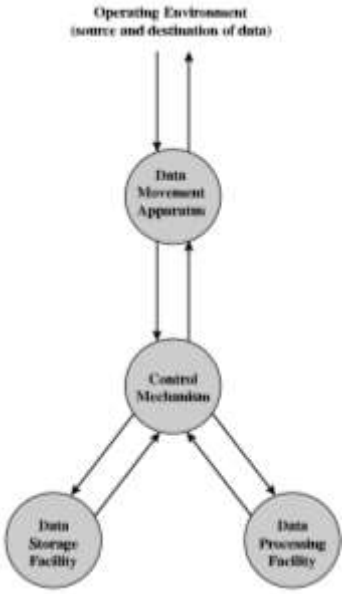
Interrupt: short IO wait



Interrupt: long IO wait

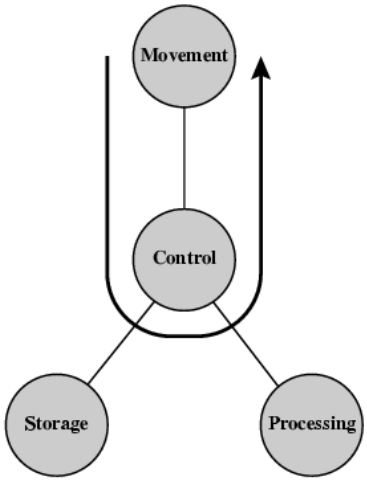
Lecture 11: Functional-organization

Q4.3: compare between Data movement, Storage, Processing from/to storage, and processing from storage to I/O, by drawing a line that shows data flow.

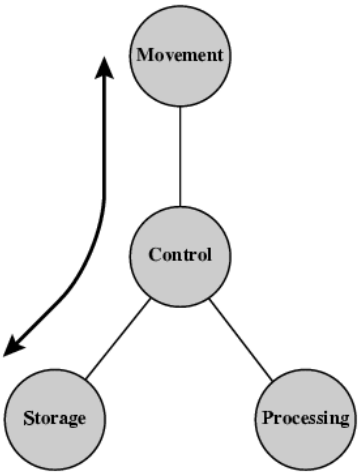


Answer

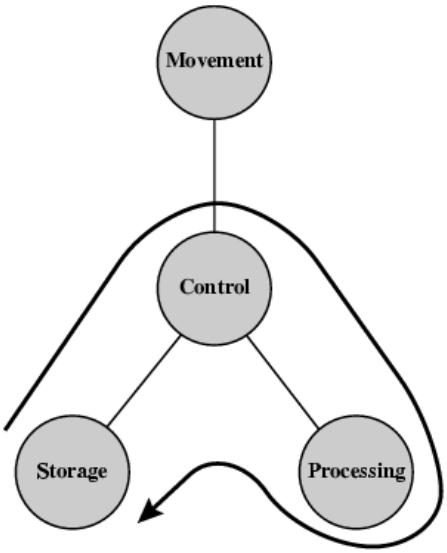
Data movement



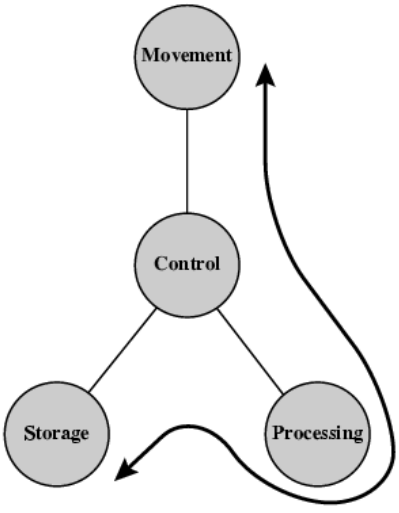
Storage



Processing from/to storage



Processing from storage to I/O



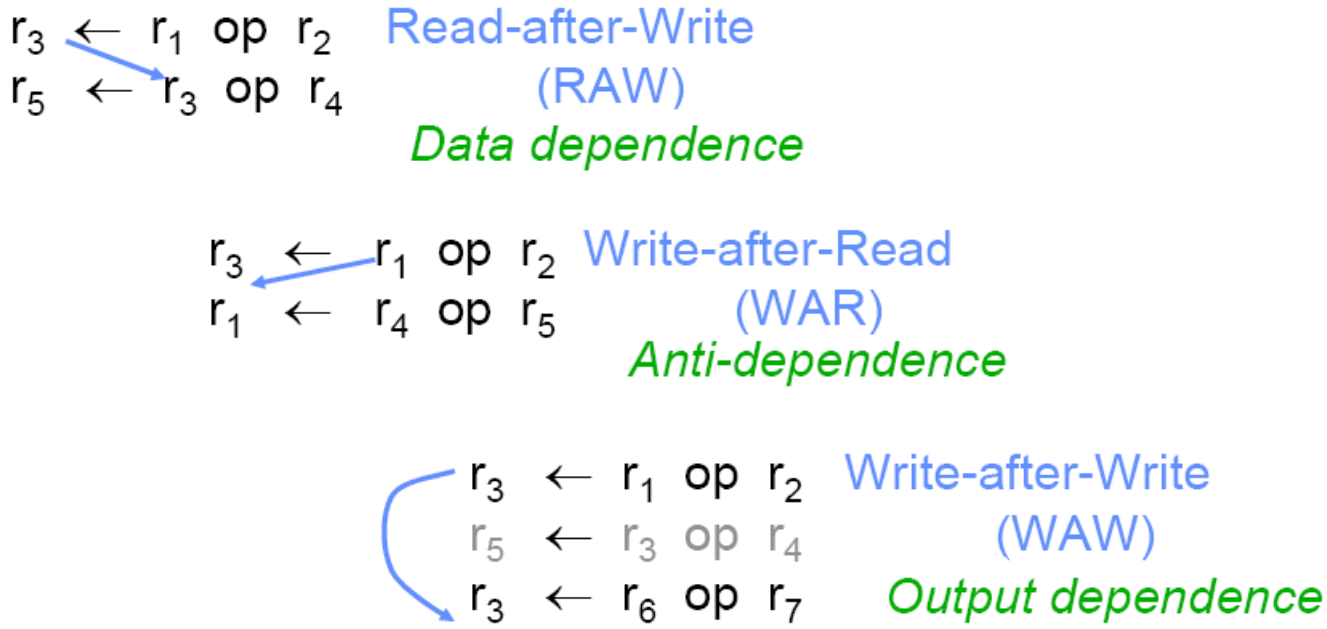
Q4.4: what is the hazard in each of the following set of operations?

$r_3 \leftarrow r_1$ op r_2
$r_5 \leftarrow r_3$ op r_4

$r_3 \leftarrow r_1$ op r_2
$r_1 \leftarrow r_4$ op r_5

$r_3 \leftarrow r_1$ op r_2
$r_5 \leftarrow r_3$ op r_4
$r_3 \leftarrow r_6$ op r_7

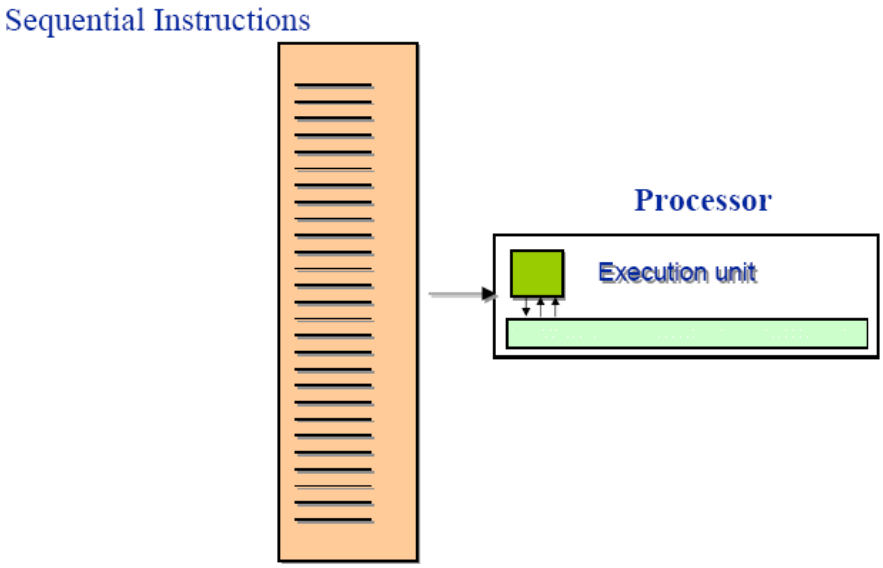
Answer



Q5: draw the 3 ILP Architecture types

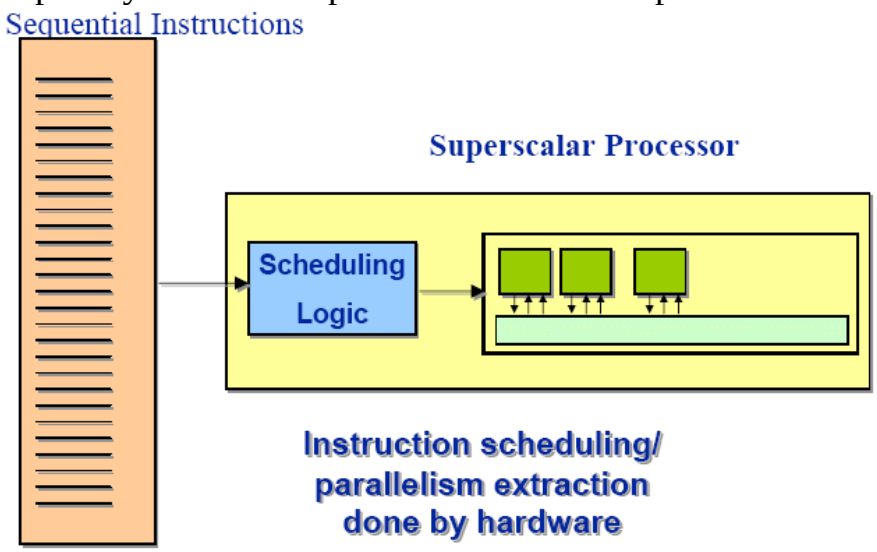
1-Sequential Architectures

– The program is not expected to convey any explicit information regarding parallelism



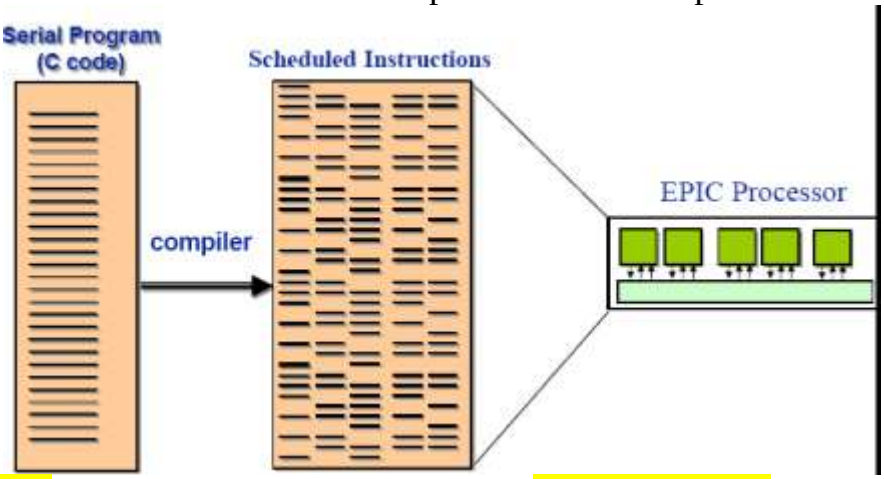
2- Dependence Architectures

– The program explicitly indicates dependencies between operations



3-Independence Architectures

Program provides information as to which operations are independent of one another

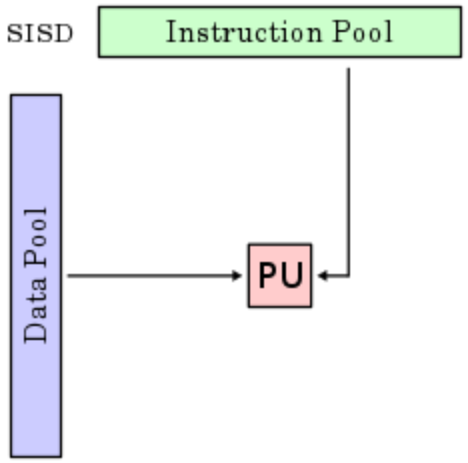


Q5.1 VLIW/EPIC processors are examples of Independence architectures

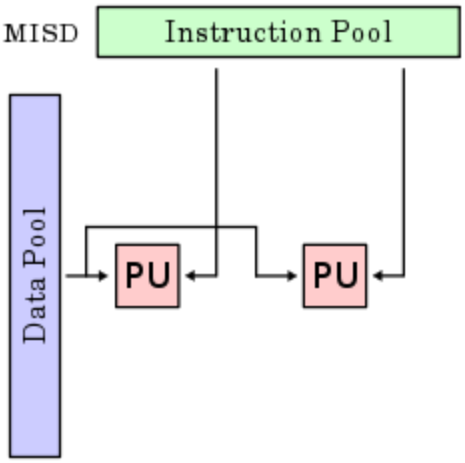
Lecture 12: Multiprocessor architectures

Q6: draw the Flynn's taxonomy of computer architectures.

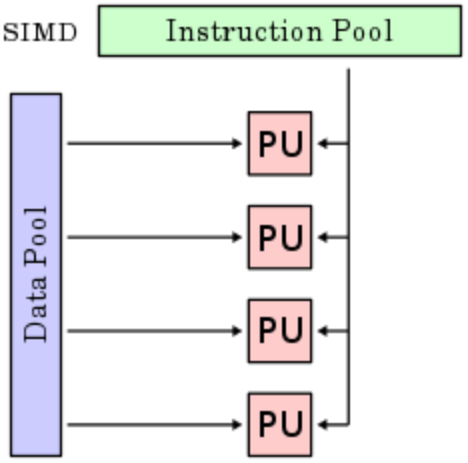
SISD



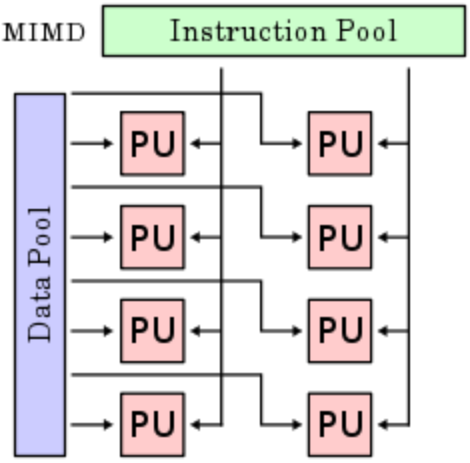
MISD



SIMD



MIMD



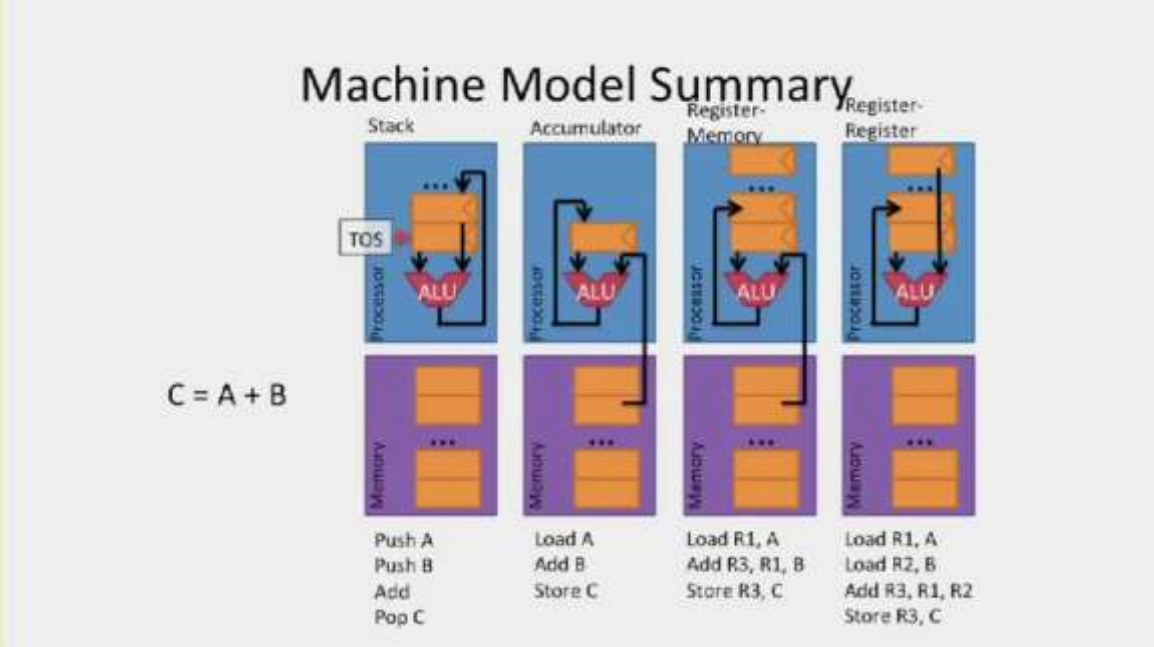
Q7: compare between CISC and RISC?

يجيب الطالب بأسلوبه

Lecture 13: Performance enhancements

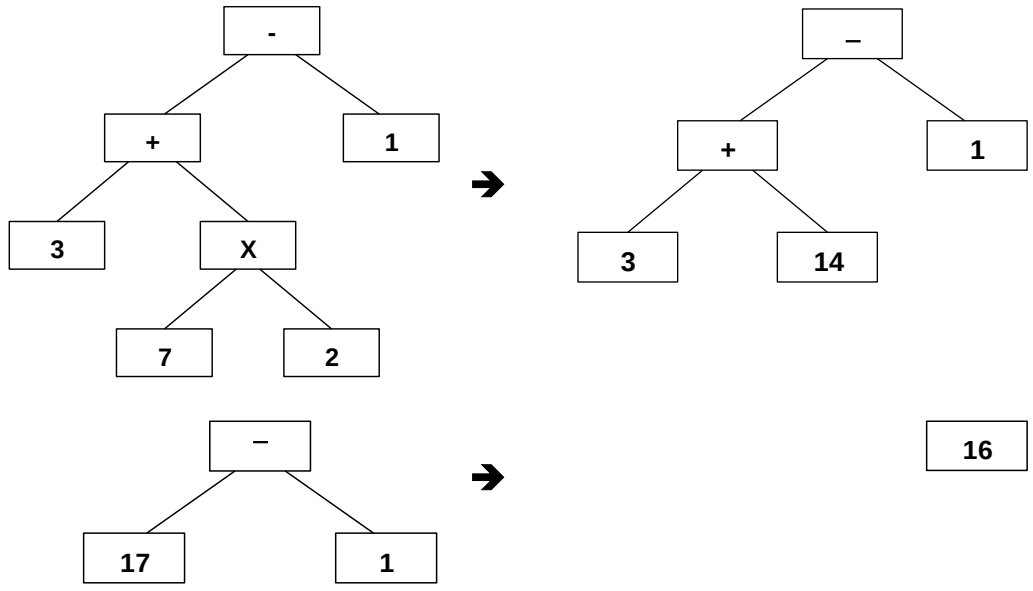
Q8: compare the memory models in calculating C=A+B

Answer



Q9: Convert the expression 3 + 7 × 2 – 1 into Prefix and evaluate using Tree

Expression tree for 3 + 7 × 2 – 1 = - + 3 × 7 2 1



Q9.1: Create an expression tree for the following expression: 6 * 4 / 2 + 7 - 5 * 3

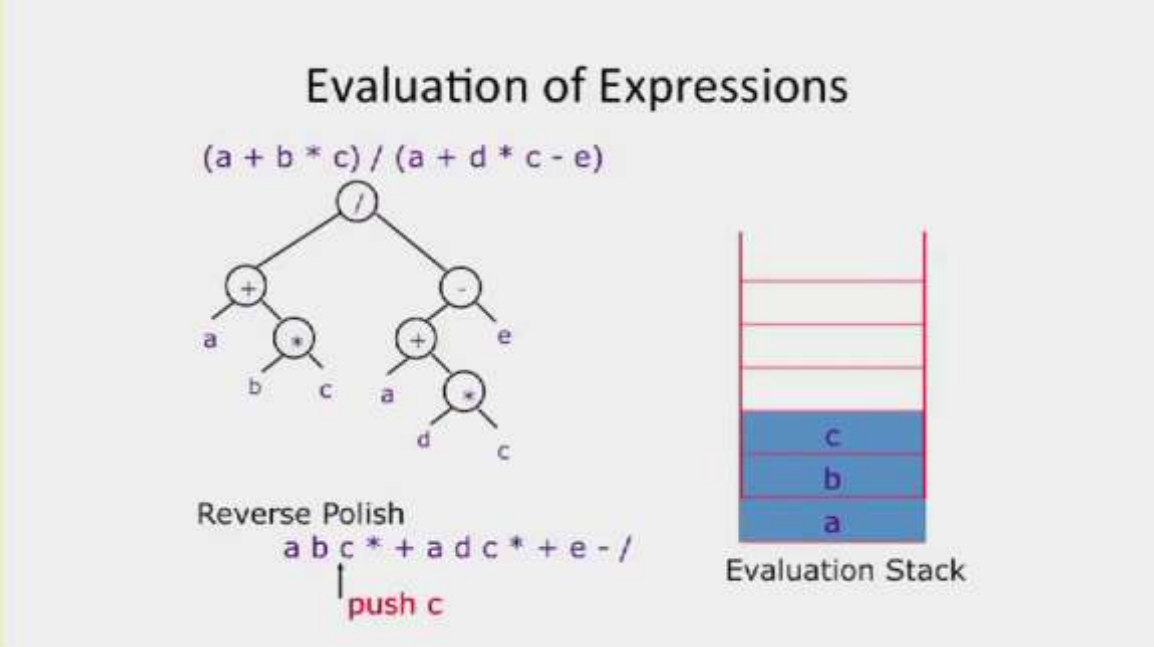
Q10: Convert the expression (A+B)*C into Postfix and evaluate using stack

(A+B)*C AB+C*
infix postfix

Enter values for three variables A=1, B=2 and C=3.

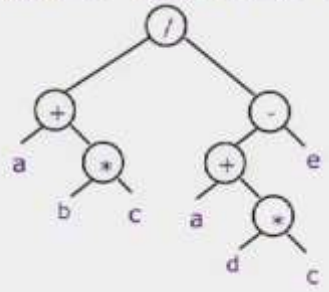
AB+C*	AB+C*	AB+C*	AB+C*	AB+C*
1	2 1	3	3 3	9
push A	push B	pop B pop A push A+B	push C	pop C pop A+B push (A+B)*C

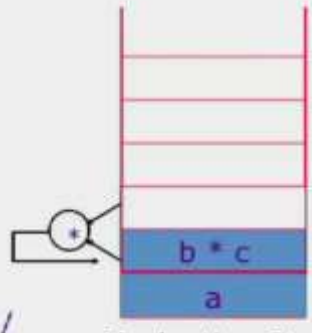
Q10. 1: evaluate expression (a+b*c)/(a+d*c-e) using stack



Evaluation of Expressions

(a + b * c) / (a + d * c - e)





Evaluation Stack

Reverse Polish

a b c * + a d c * + e - /

multiply

Q11: مسألة

"Iron Law" of Processor Performance

$$\frac{\text{Time}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} * \frac{\text{Cycles}}{\text{Instruction}} * \frac{\text{Time}}{\text{Cycle}}$$

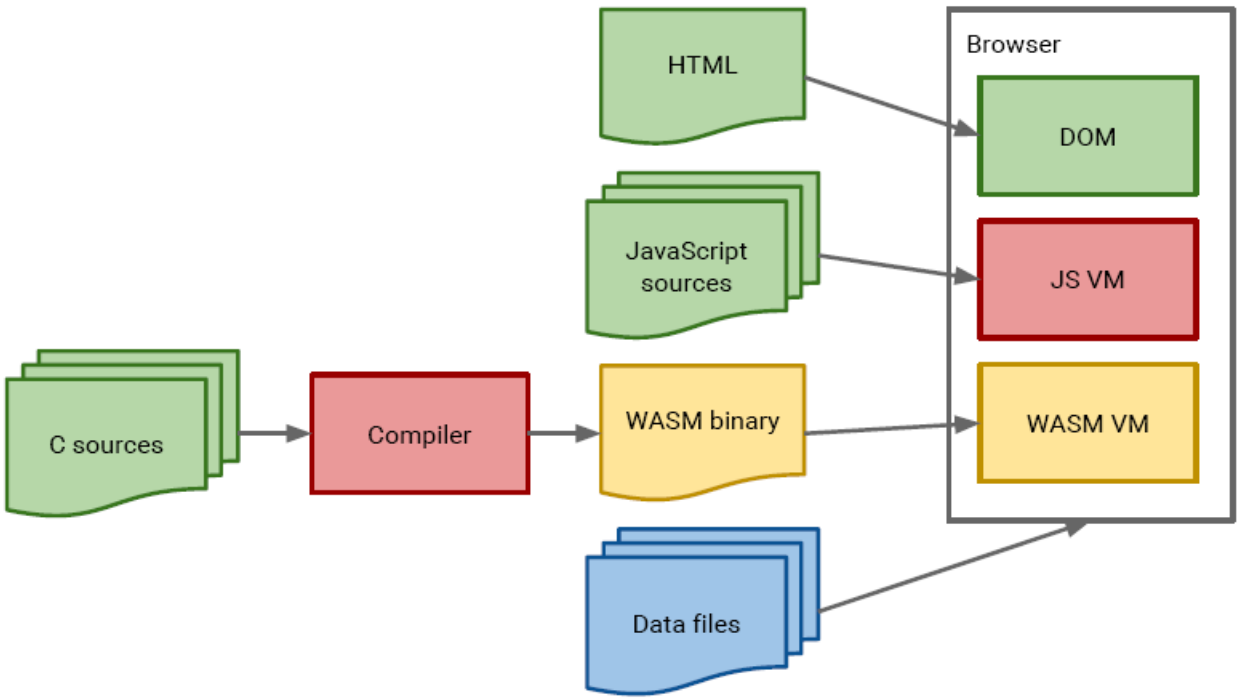
- Instructions per program depends on source code, compiler technology, and ISA
- Cycles per instructions (CPI) depends upon the ISA and the microarchitecture
- Time per cycle depends upon the microarchitecture and the base technology

Microarchitecture	CPI	cycle time
Microcoded	>1	short
Single-cycle unpipelined	1	long
Pipelined	1	short

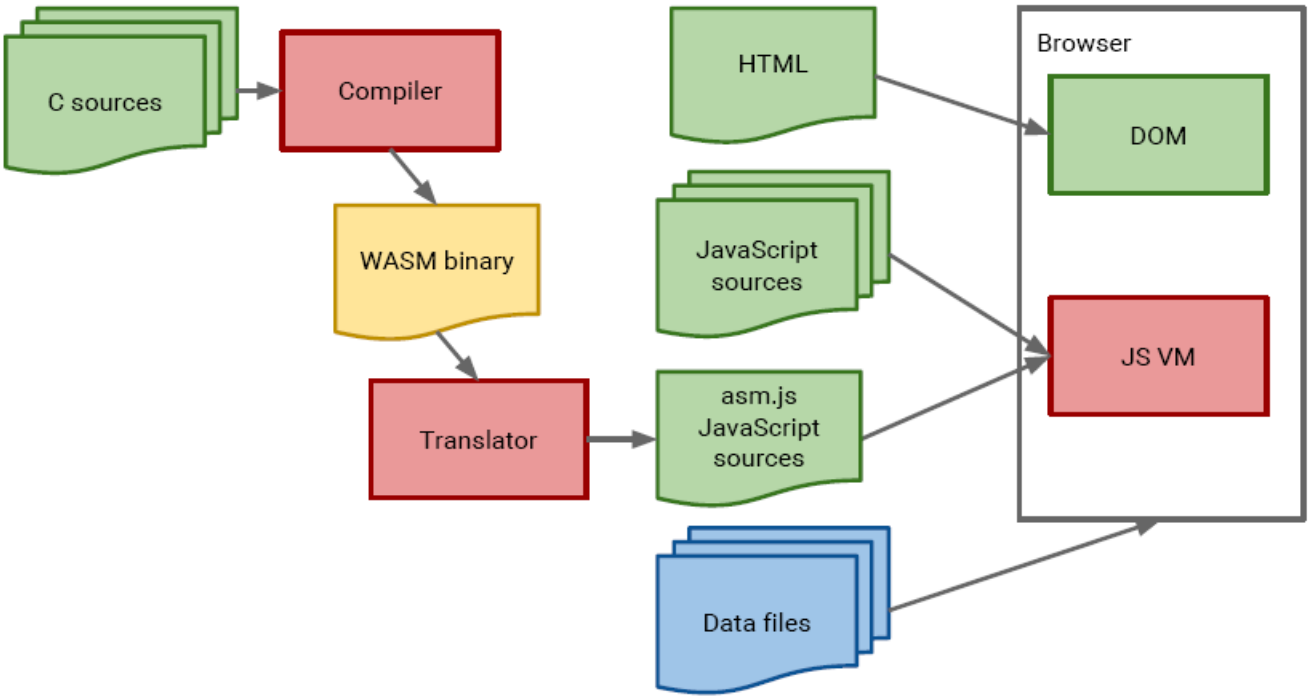
Lecture 14: Modern architectures

Q12: draw the architecture of WASM(web assembly) and WASM Polyfill.

WASM



WASM Polyfill



Lecture 8: Mid-term and Intoduction to Assembly

منهج الدكتورة أمل
نصف المنهج