Multi-Thread
Parallelism

Outline

* Some Examples
  - CM*
  - HEP
  - Cosmic Cube
* MP vs. Multicomputer Network
* One Supercomputer vs. "The Multi"
* Amdahl's Law
* Speed-up, Efficiency, Utilization, Redundancy
* Interconnection Networks
* CACHE COHERENCY
* SEQUENTIAL CONSISTENCY

Tightly-coupled vs Loosely-coupled

Tightly coupled (i.e., Multiprocessor)

DSM

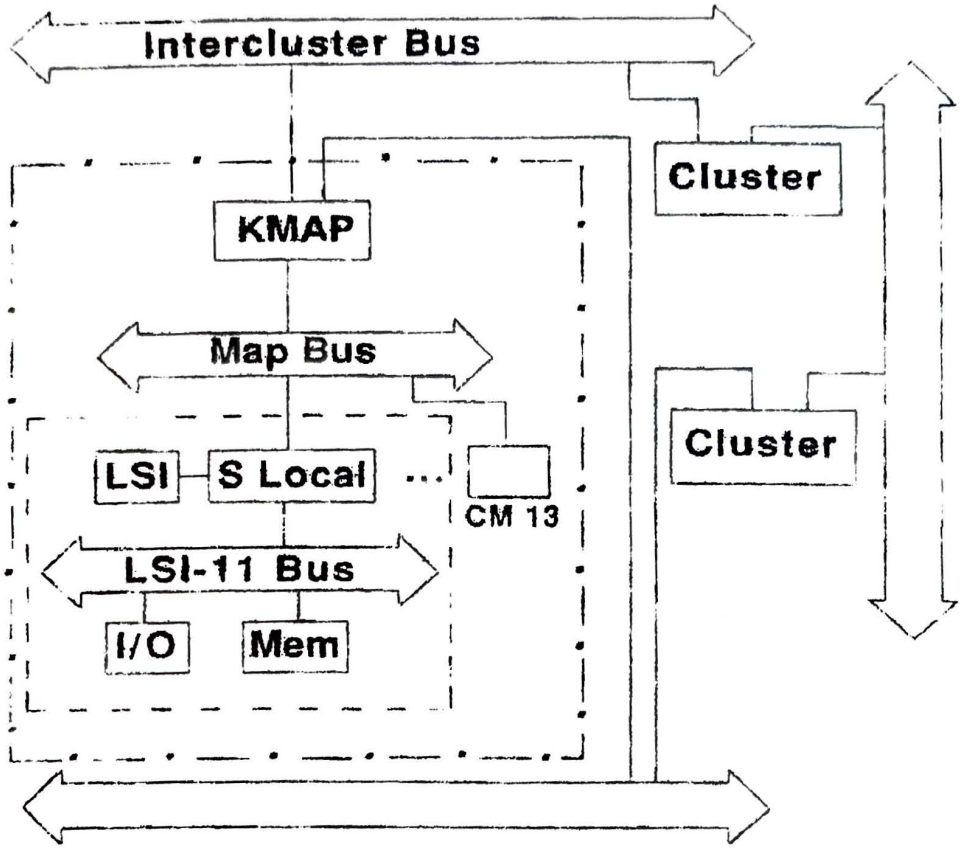
- Shared memory
- Each processor capable of doing work on its own
- Easier for the software
- Hardware has to worry about cache coherency, memory contention

Coprocessor

Loosely-coupled (i.e., Multicomputer Network)

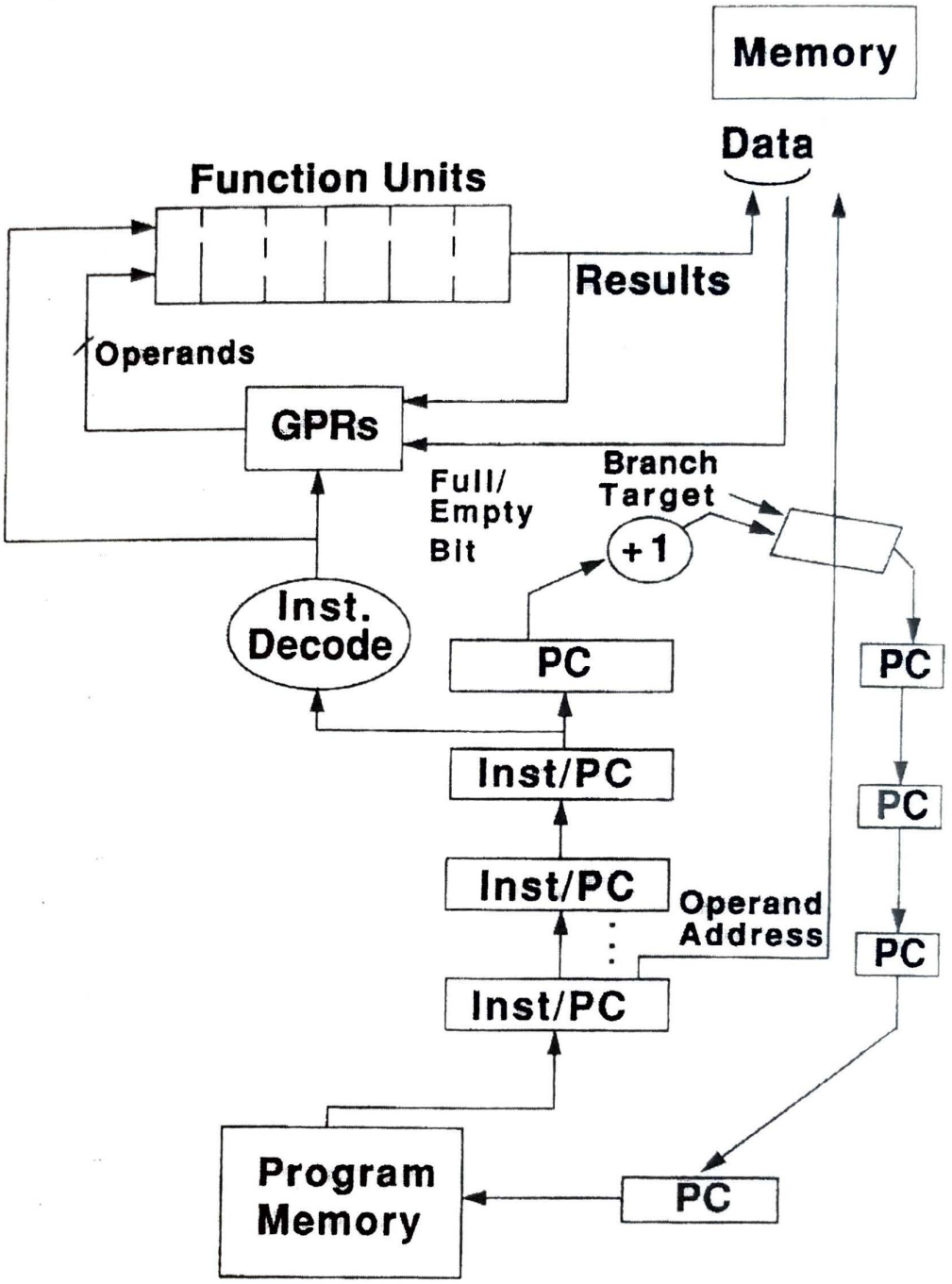
- Message passing
- Easier for the hardware
- Programmer's job is tougher

cm*



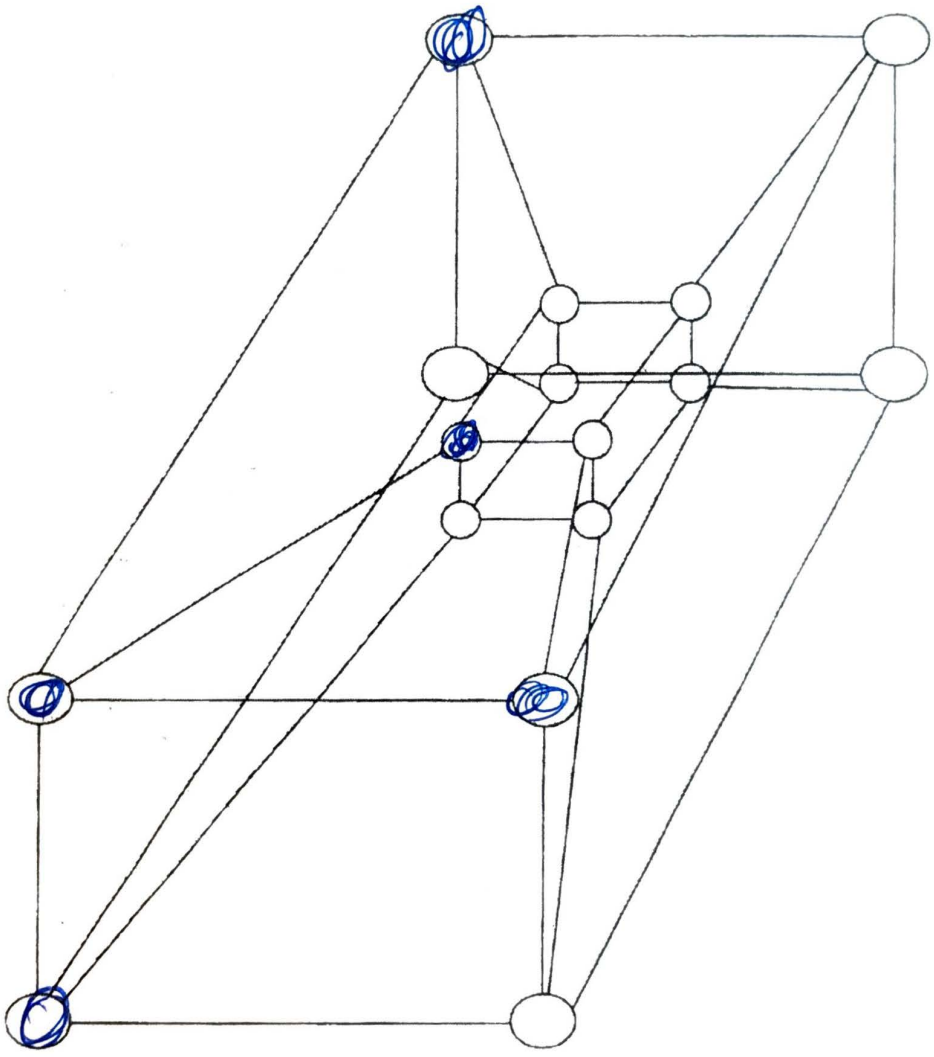
Note: ***A well-meaning student told me to get rid of this slide. cm* is old. People will think you are an old man, and not take you seriously.***

The HEP


Memory
Data
Function Units
Results
Operands
GPRs
Full/
Empty
Bit
Branch
Target
+1
Inst.
Decode
PC
PC
Inst/PC
PC
Inst/PC
Operand
Address
PC
Inst/PC
Program
Memory
PC

Cosmic Cube

(Example: k = 4)



One Supercomputer

vs.

"The Multi"

(…Except <u>Even</u> Supercomputers have adopted the multi approach)

$$1 * 2^n \qquad 2^k * 2^{n-k} \qquad 2^n * 1$$

Why do we care?

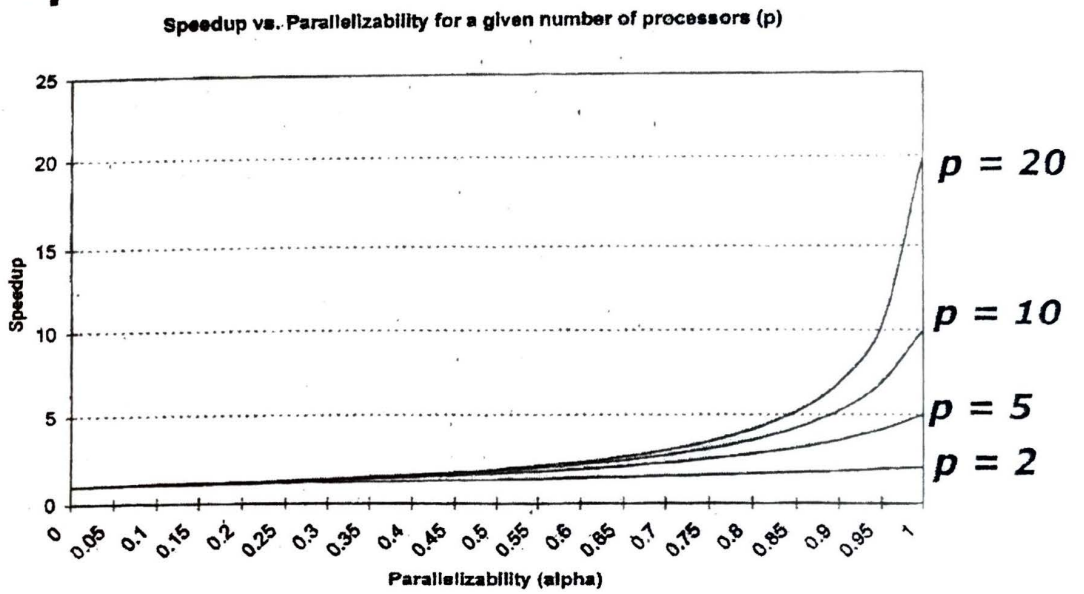
- ***-- Economic Answer***
- ***-- Strategic Answer***
- ***-- Scientific Answer***

<u>Scalability</u>

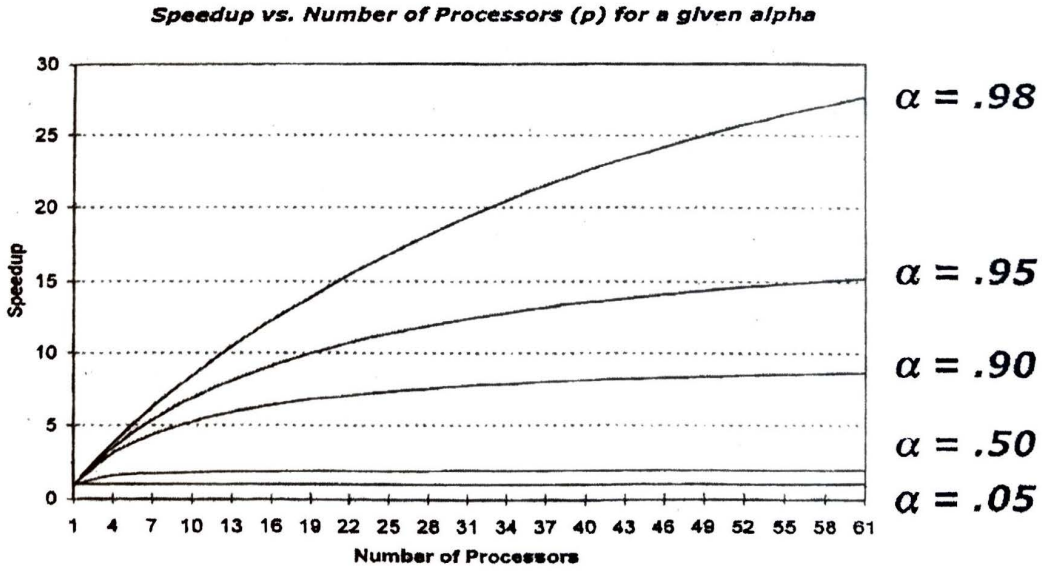
- SIMD easy
- MIMD hard
- Very large Scale
  - CM1 - Thinking Machines 2^{16} cores
  - non-Von - 2^{20} cores
  - Boolean Vector Machine - 2^{30} cores

Amdahl's Law

* **Speed-up as a function of the parallelizability (α) of the application**



* **Speed-up of an application as we add more and more processors (p)**



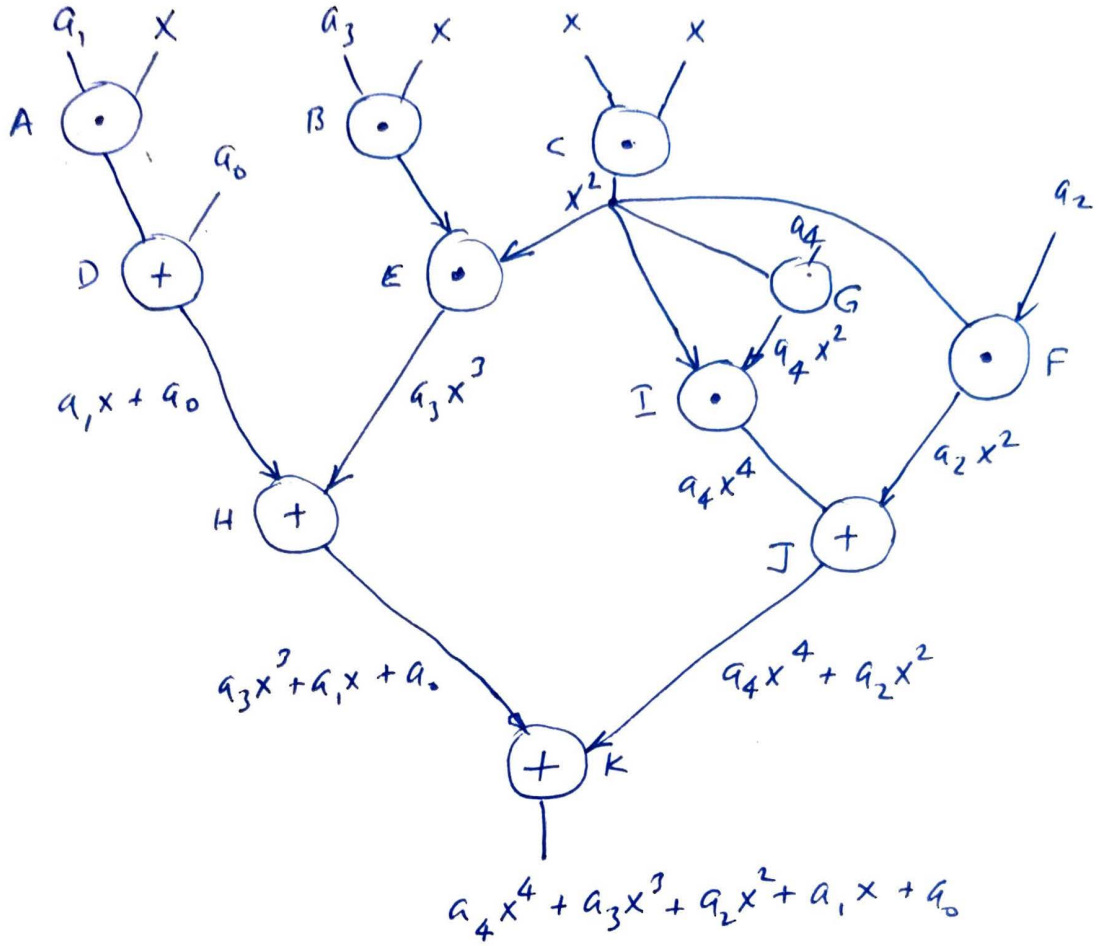
$$\boxed{T_p} = \frac{\alpha \boxed{T_1}}{p} + \frac{(1-\alpha)T_1}{1}$$

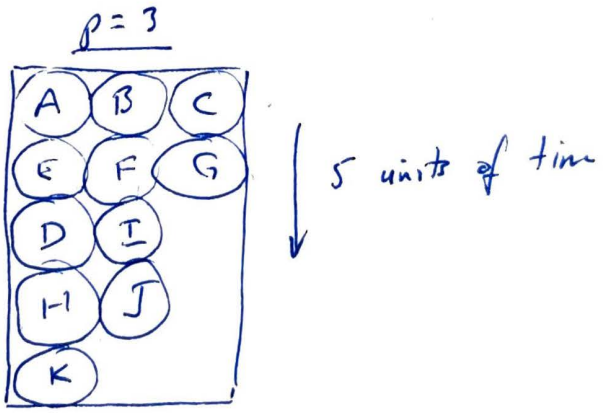
$$S_p = \frac{T_1}{T_p} = \boxed{\frac{\cancel{T_1}}{\alpha \frac{\cancel{T_1}}{p} + (1-\alpha)\cancel{T_1}}}$$

$$\boxed{S_p = \frac{1}{\frac{\alpha}{p} + (1-\alpha)}}$$

MP/6

EXAMPLE : $a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x^1 + a_0$

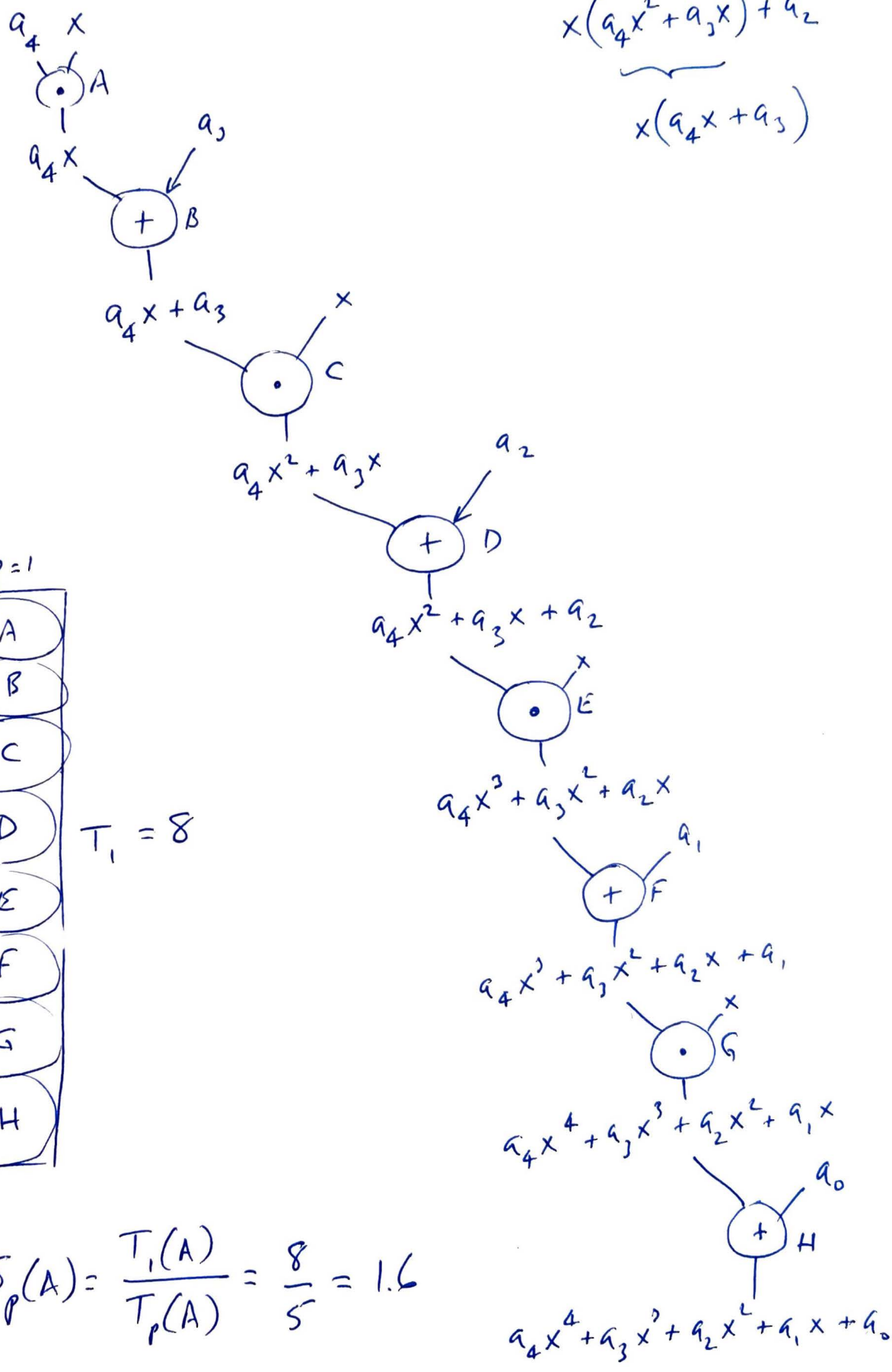


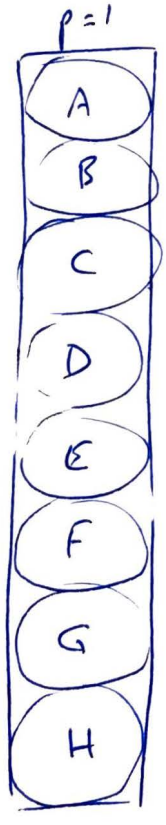


$$S_p(A) = \frac{T_1(A)}{T_p(A)} = \frac{11}{5} = 2.2$$

Right? $\boxed{\text{WRONG}}$ Why?

Note: $a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = x\big(\underbrace{a_4x^3 + a_3x^2 + a_2x^1}_{x(\underbrace{a_4x^2 + a_3x}_{x(a_4x + a_3)}) + a_2} + a_1\big) + a_0$

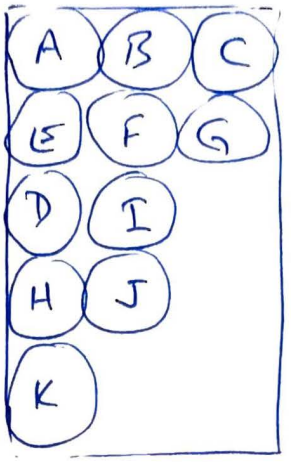




$T_1 = 8$

$$S_p(A) = \frac{T_1(A)}{T_p(A)} = \frac{8}{5} = 1.6$$

$p = 3$



$$\text{Efficiency} = \frac{1 \cdot T_1}{p \cdot T_p} = \frac{8}{3 \times 5} = \frac{8}{15}$$

$$\text{Utilization} = \frac{O_p}{p \cdot T_p} = \frac{11}{3 \cdot 5} = \frac{11}{15}$$

$$\text{Redundancy} = \frac{O_p}{O_1} = \frac{11}{8}$$

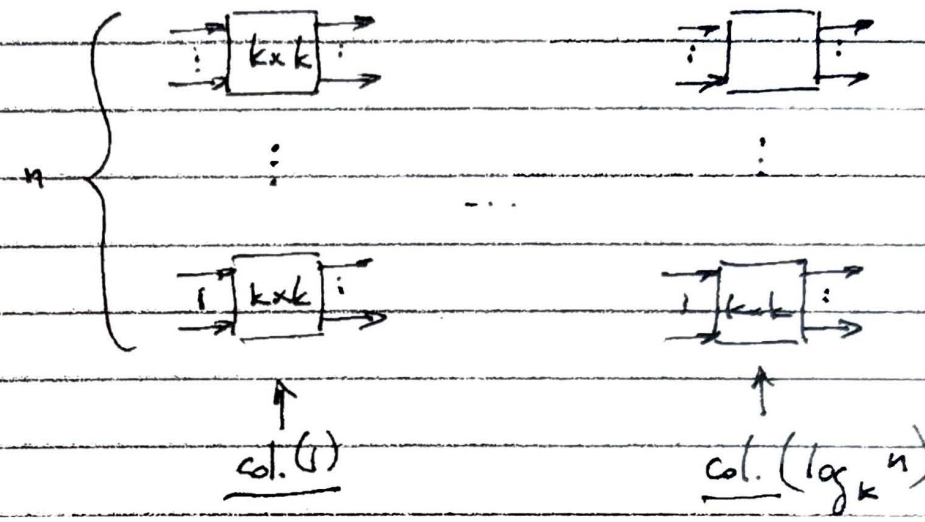
$$\boxed{E \cdot R = U}$$

INTERCONNECTION NETWORKS

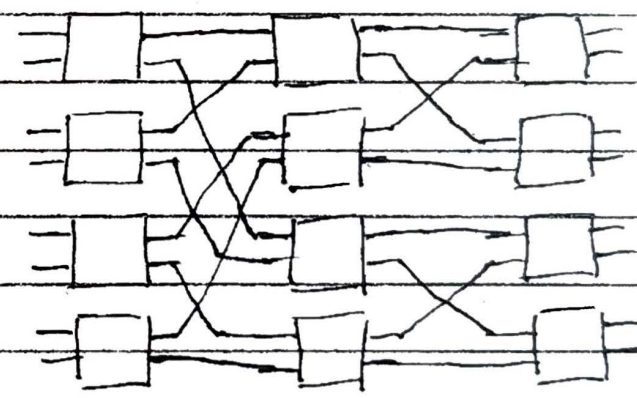
Type	Cost	Latency	Contention
Bus	$O(n)$	1	Worst
Full crossbar	$O(n^2)$	1	Best
Omega Network	$nk \log_k n$	$O(\log_k n)$	
TREE	$O(n)$	$O(\log_2 n)$	
HYPERCUBE	$O(n \log n)$	$O(\log n)$	
RING	$O(n)$	$O(n)$	
MESH	$O(n)$	$O(\sqrt{n})$	

MORE DETAIL, OMEGA NETWORK

* OMEGA NETWORK (DUNCAN LAURIE, UIUC)



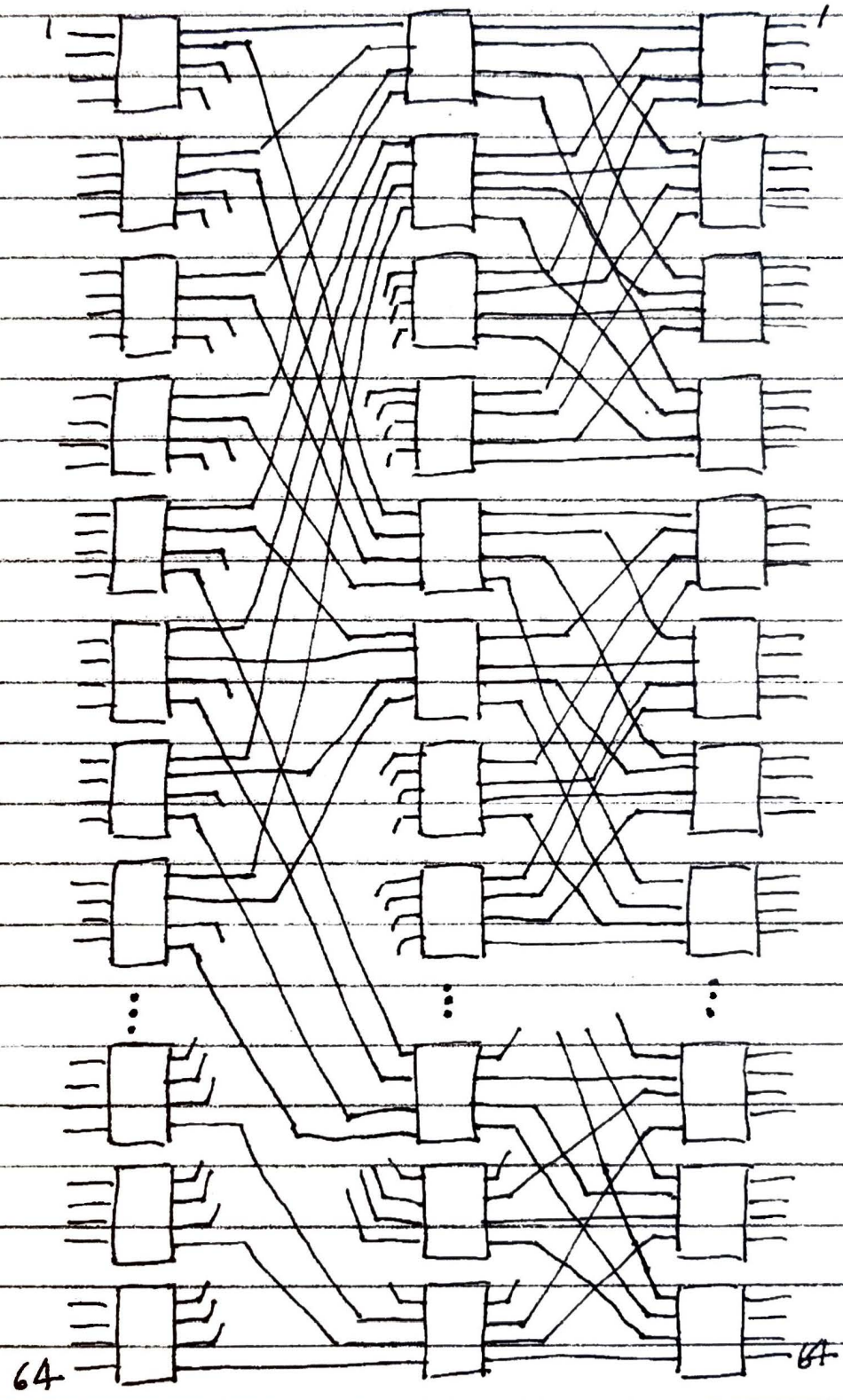
EXAMPLE : $n = 8$, $k = 2$



* BANYAN TREE (G. JACK LIPOVSKI, UT)

l = # of levels of switches
p = # of connections on processor side
m = # of connections on memory side

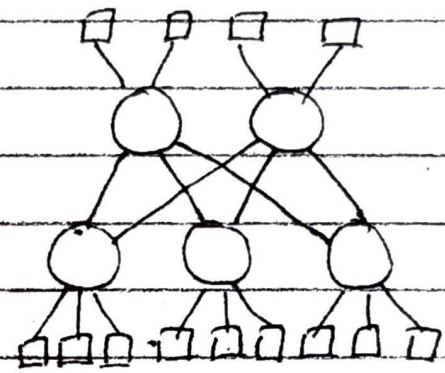
Omega Networks (Bonus Example)



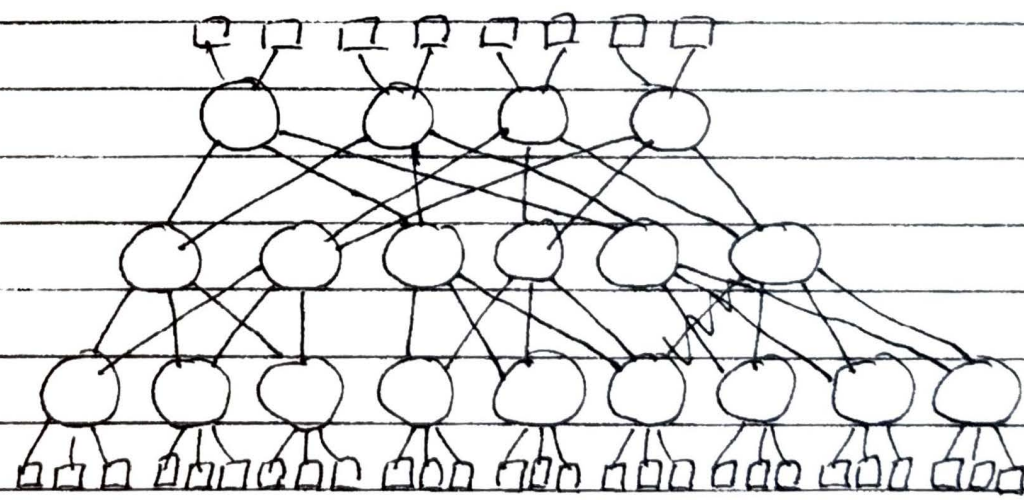
BANYAN TREE (CONTINUED)

A SWITCH :

EXAMPLE: $l = 2$, $p = 2$, $m = 3$

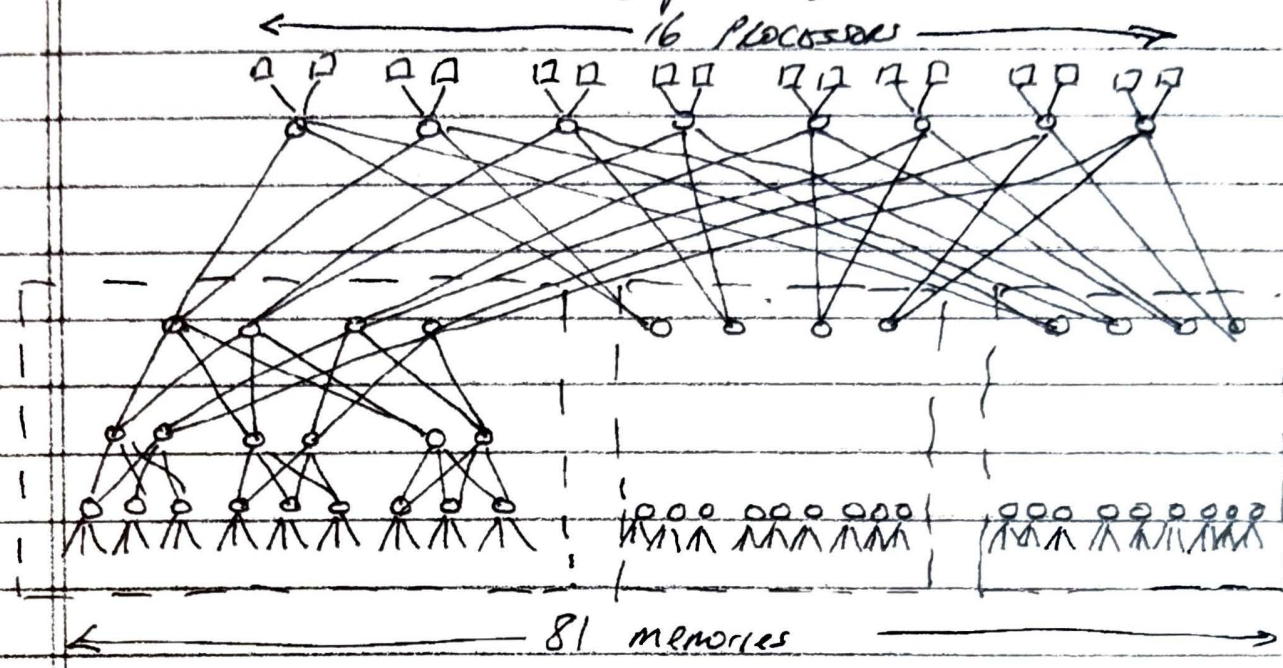


EXAMPLE : $l = 3$, $p = 2$, $m = 3$



BANYAN TREE (CONT.)

EXAMPLE: $l = 4$, $p = 2$, $m = 3$



NOTE: p^l PROCESSORS, m^l MEMORIES

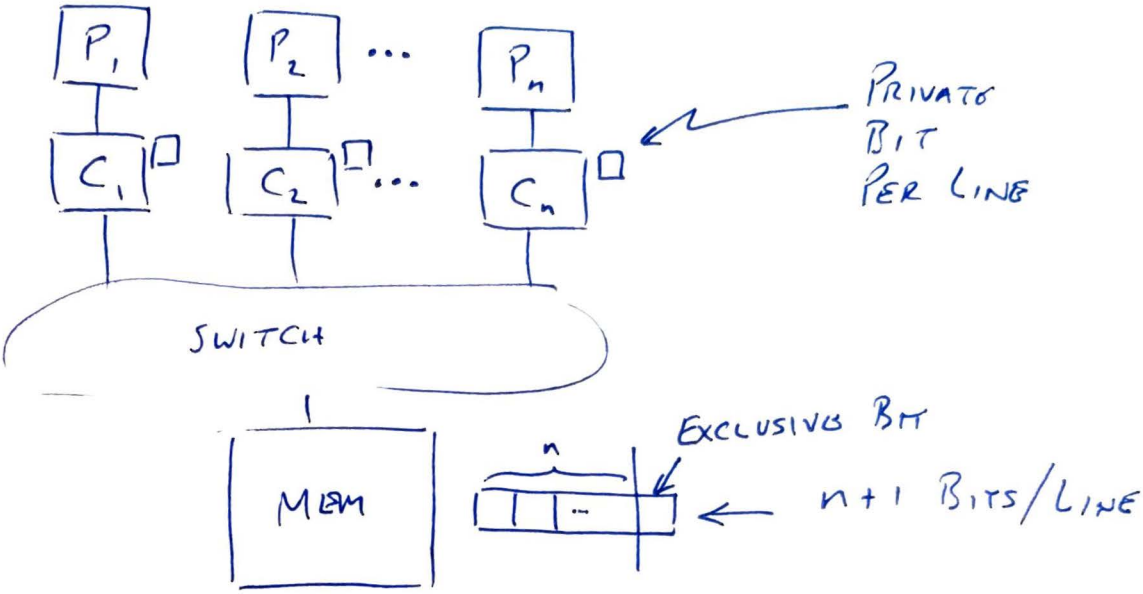
Level 1 SWITCHES : p^{l-1}
" 2 " : $m \cdot p^{l-2}$
" 3 " : $m^2 \cdot p^{l-3}$
⋮
" l " : m^{l-1}

NOTE: FOR BOTH OMEGA AND BANYAN:

A UNIQUE PATH FROM EACH TO EACH.

CACHE COHERENCY

DIRECTORY



MEMORY DIRECTORY CONSISTS OF $n+1$ bits/CACHE LINE

* IF CACHE HAS IT ~~WITH~~ IN READ MODE, BIT = 1
* IF CACHE HAS IT IN WRITE MODE, EXCLUSIVE BIT = 1

IN PRIVATE CACHE, IF PRIVATE BIT = 1, WRITE MODE

* PRIVATE CACHES ARE NOT WRITE THROUGH

EXAMPLES FOR $n = 3$:

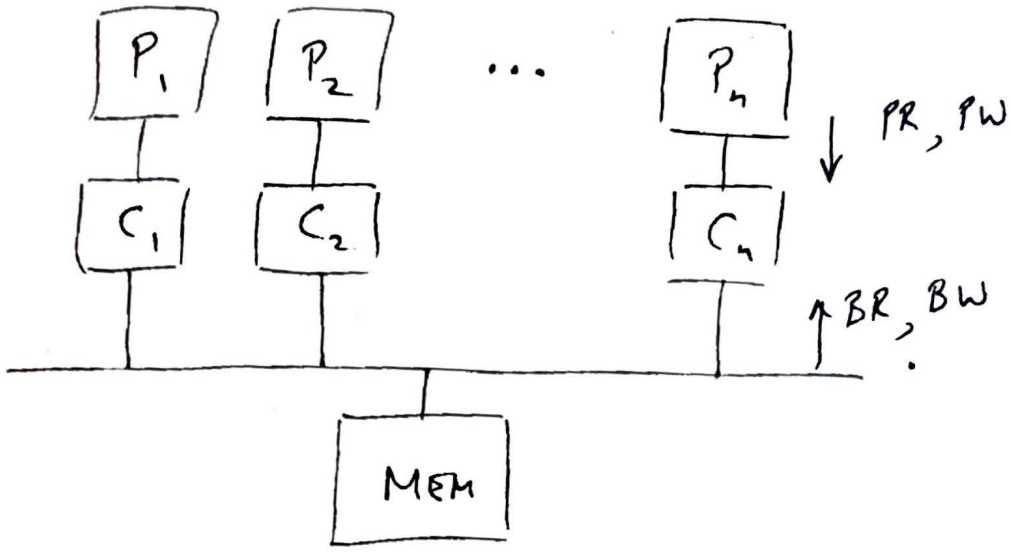
1	0	1	0

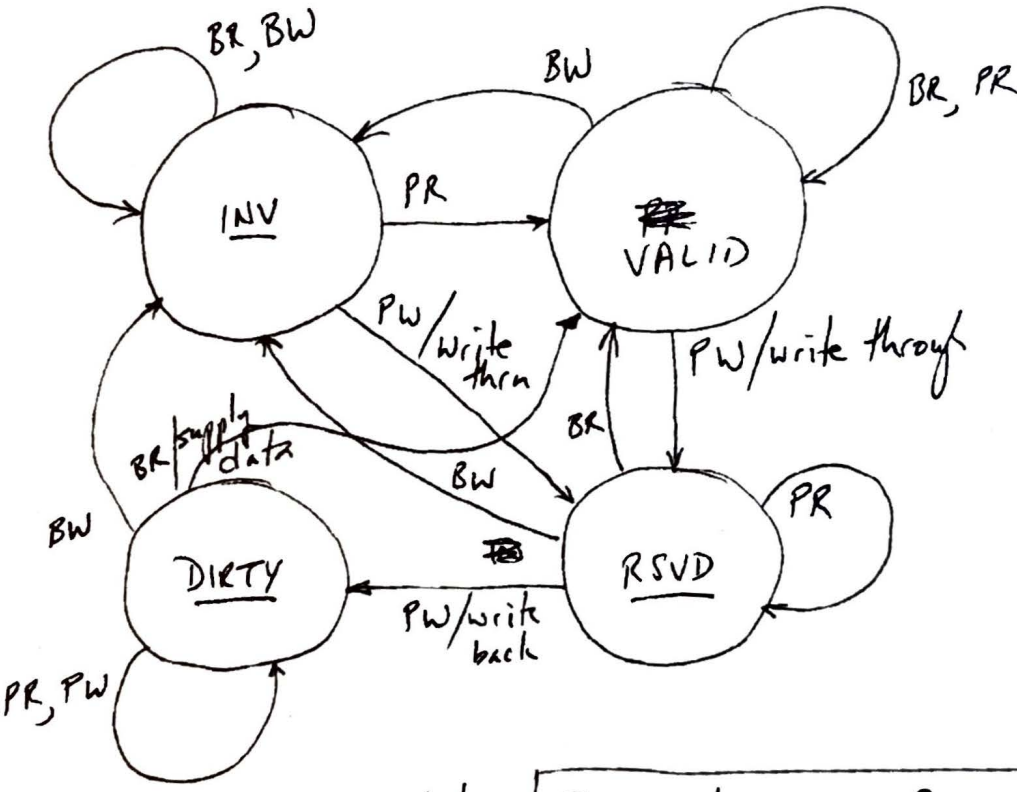
C_1, C_3 HAVE IT IN READ MODE

0	1	0	1

C_2 HAS IT IN WRITE MODE

SNOOPY CACHE

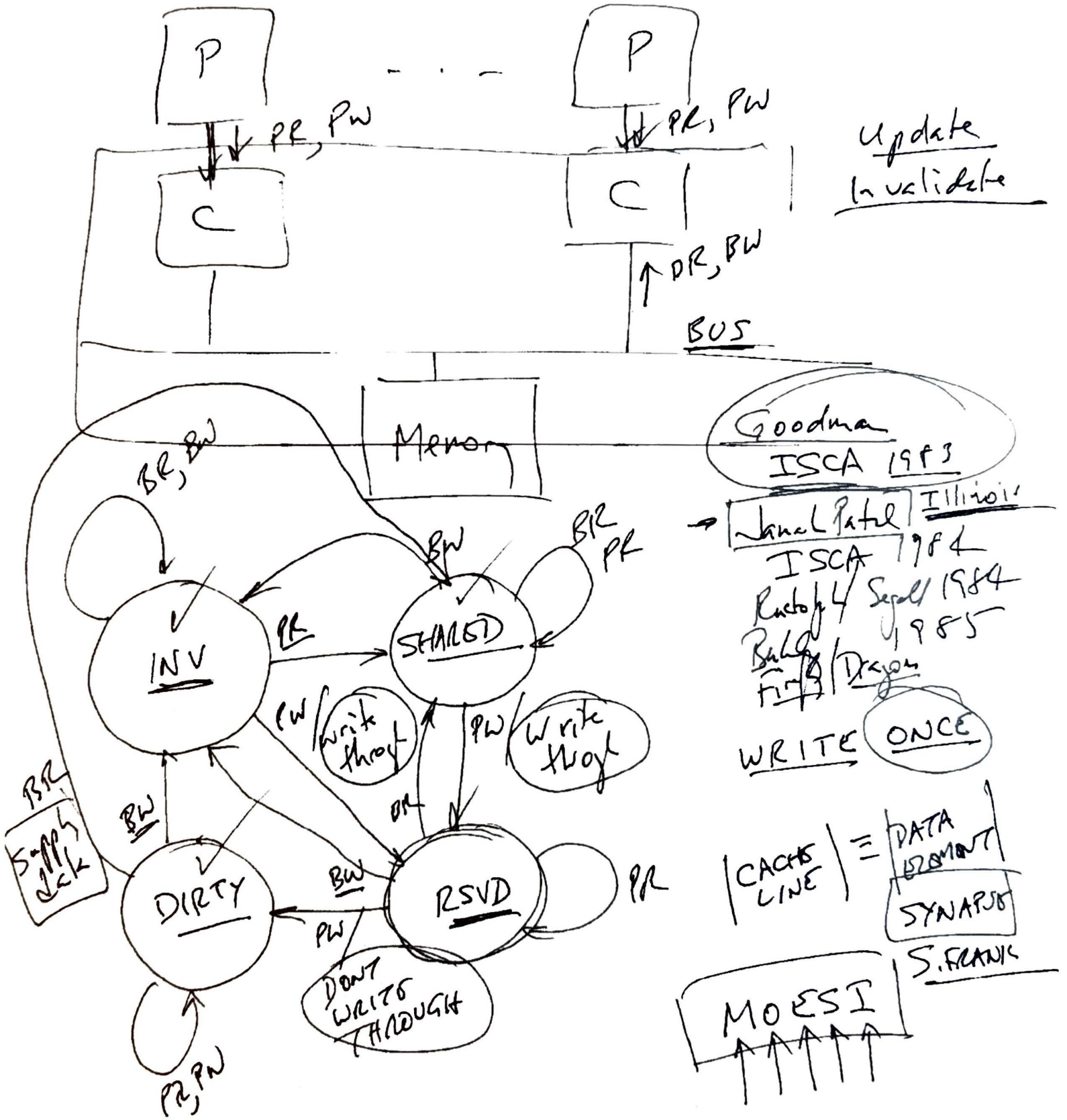




Note: CACHE LINE = ONE ELEMENT

MESI / MOESI /

Cache Coherency
(Snoopy Cache)



SEQUENTIAL CONSISTENCY

WHY: TO PROTECT AGAINST TWO THREADS SIMULTANEOUSLY ACCESSING A CRITICAL SECTION

	THREAD 1		THREAD 2
	$L1 = \phi$		$L2 = \phi$
	⋮		⋮
A	$L1 \leftarrow 1$	X	$L2 \leftarrow 1$
B	if $(L2 = \phi)$	Y	if $(L1 = \phi)$
	{critical section		{critical section}
C	$L1 \leftarrow \phi$	Z	$L2 \leftarrow \phi$

WHAT CAUSES THE PROBLEM:

$$B \rightarrow X \quad \text{AND} \quad Y \rightarrow A$$

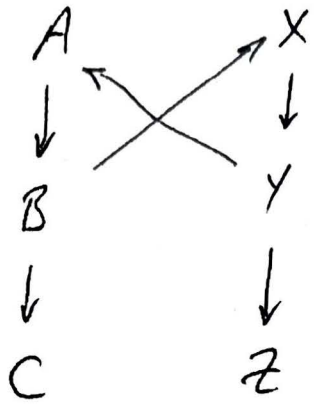
SEQUENTIAL CONSISTENCY

LD/ST ACCESS MEMORY IN PROGRAM ORDER.

i.e. $A \rightarrow B \rightarrow C$ AND

$X \rightarrow Y \rightarrow Z$

IN ORDER FOR CRITICAL SECTION TO BE ACCESSIBLE BY BOTH AND SEQUENTIAL CONSISTENCY TO BE MAINTAINED



$\Rightarrow A \rightarrow B \rightarrow X \rightarrow Y \rightarrow A$

IMPOSSIBLE!

∴ SEQUENTIAL CONSISTENCY → MUTUALLY EXCLUSIVE ACCESS TO CRITICAL SECTION

ACCESSES (LEGITIMATE) FOR SEQUENTIAL CONSISTENCY

A	A	A	A	A	A	A	A	A	A
B	B	B	B	X	X	X	X	X	X
C	X	X	X	B	B	B	Y	Y	Y
X	C	Y	Y	C	Y	Y	B	B	Z
Y	Y	C	Z	Y	C	Z	C	Z	B
Z	Z	Z	C	Z	Z	C	Z	C	C

1 then 2 | 1 | 2 | Now | 1