Intro to MP

∗ *What is it*

--Tightly coupled (shared memory)
--Loosely coupled (multicomputer network)

∗ *Why do it*

--incremental cost paradigm
--greater throughput
--shorten time to solve one problem

∗ *How*

--Master/Slave
--SMP

∗ *Example* $a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$

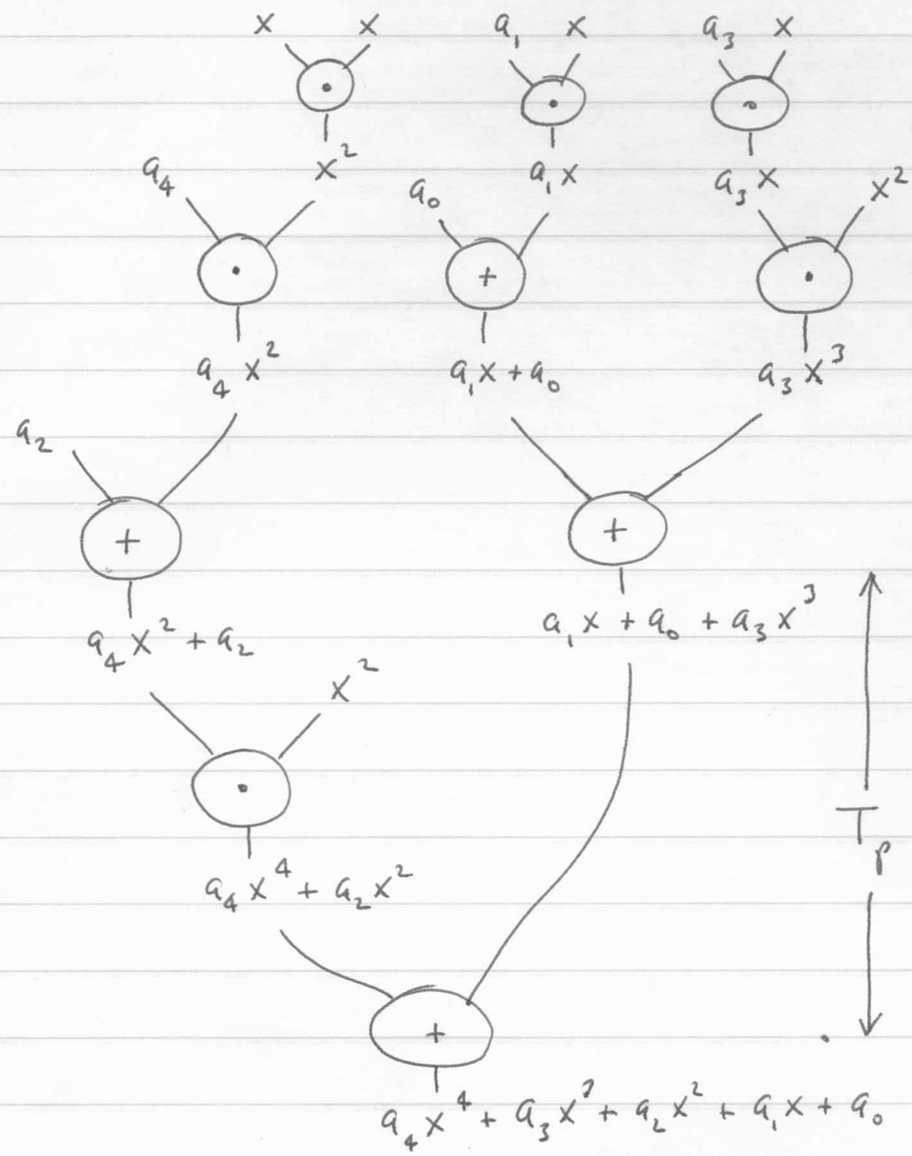
S_p, E_p, U_p, R_p

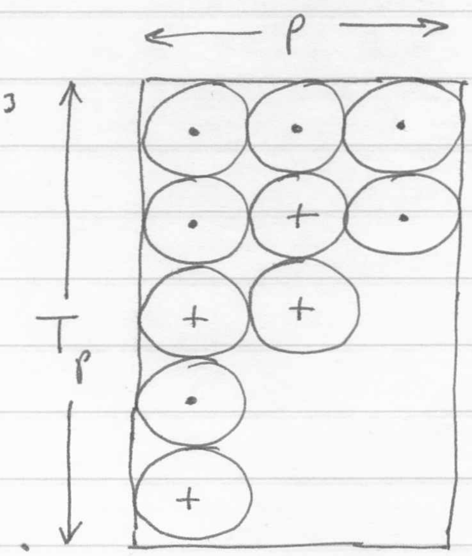
∗ Interconnection networks

-- types: bus, ~~rig~~ RING, Ω, X, tree
-- parameters: cost, contention, latency

∗ Cache consistency

$S_p,\ E_p,\ U_p,\ R_p$





Speed-up: $S_p = \dfrac{T_1}{T_p}$

Utilization: $U_p = \dfrac{O_p}{p\,T_p}$

Efficiency: $E_p = \dfrac{1 \cdot T_1}{p \cdot T_p}$

Redundancy: $R_p = \dfrac{O_p}{O_1}$

$$R_p = \frac{U_p}{E_p}$$

Amdahl's Law

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

EXAMPLE: $\alpha = 1$

$$S_p = \frac{1}{\frac{1}{p} + 1 - 1} = p$$

EXAMPLE: $\alpha = \emptyset$

$$S_p = \frac{1}{\frac{\emptyset}{p} + 1 - \emptyset} = 1.$$

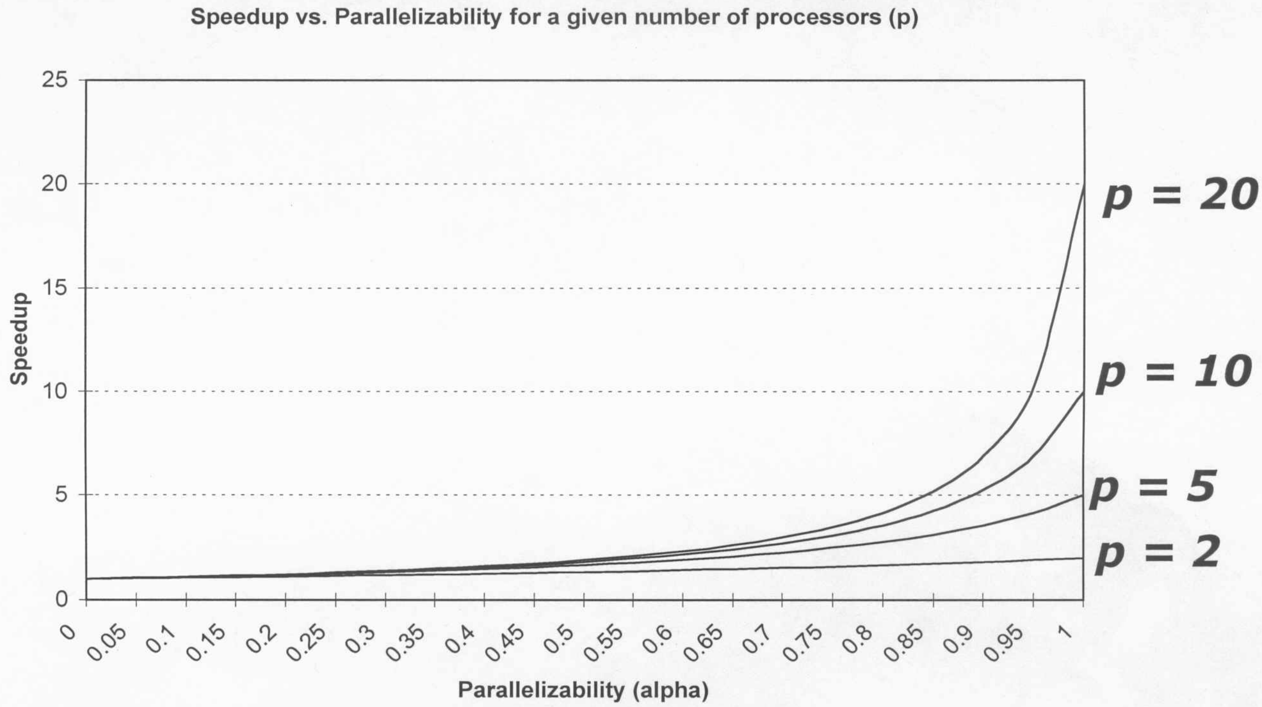
Buzbee's Observation: $T_p = \frac{\alpha T_1}{p} + (1-\alpha) T_1 + \sigma(p)$

That is,

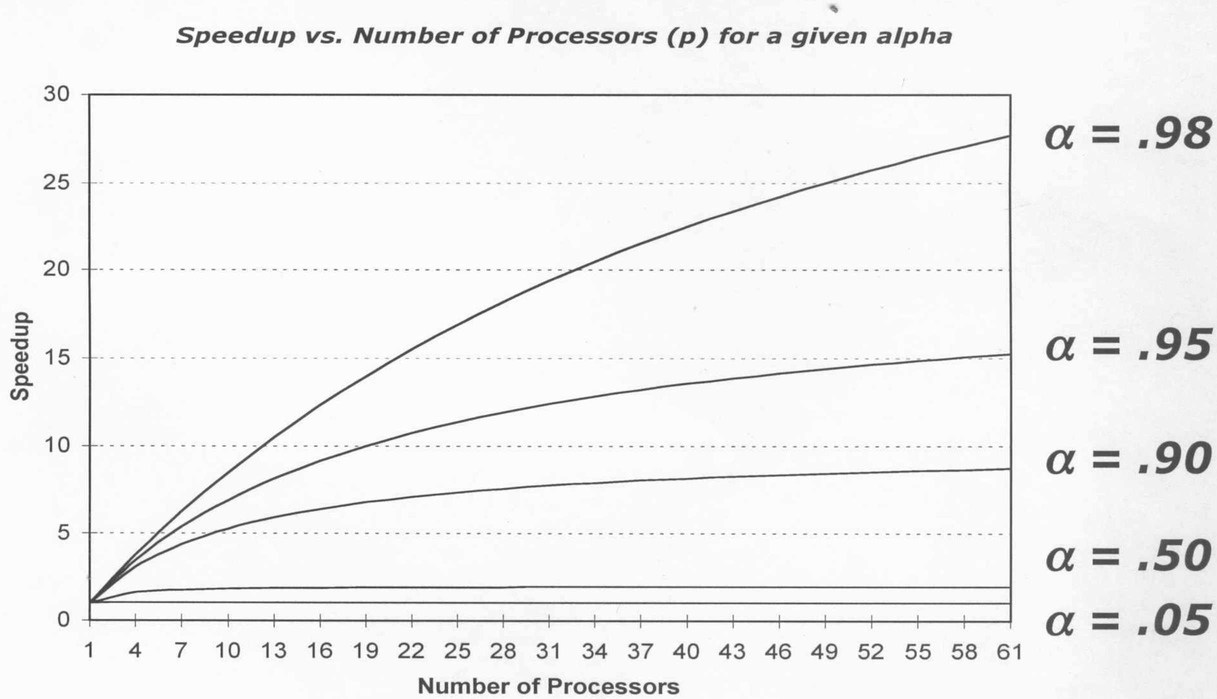
$$S_p = \frac{T_1}{T_p} = \frac{T_1}{\frac{\alpha T_1}{p} + (1-\alpha) T_1 + \sigma(p)} = \frac{1}{\frac{\alpha}{p} + (1-\alpha) + \frac{\sigma(p)}{T_1}}$$

Amdahl's Law

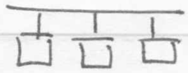
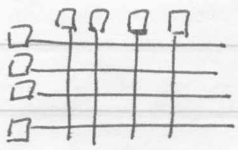
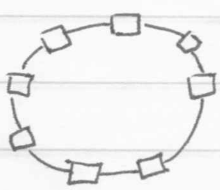
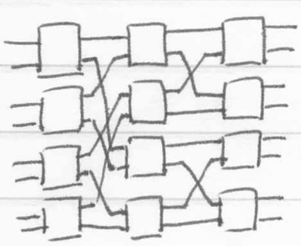
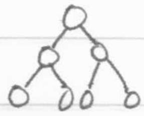
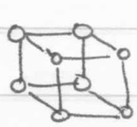
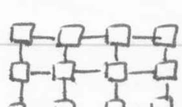
* *Speed-up as a function of the parallelizability (α) of the application*



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

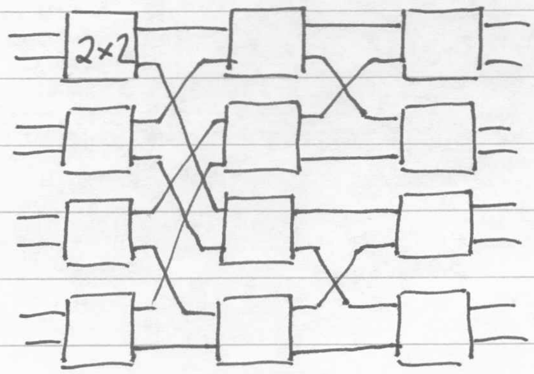


Interconnection Networks

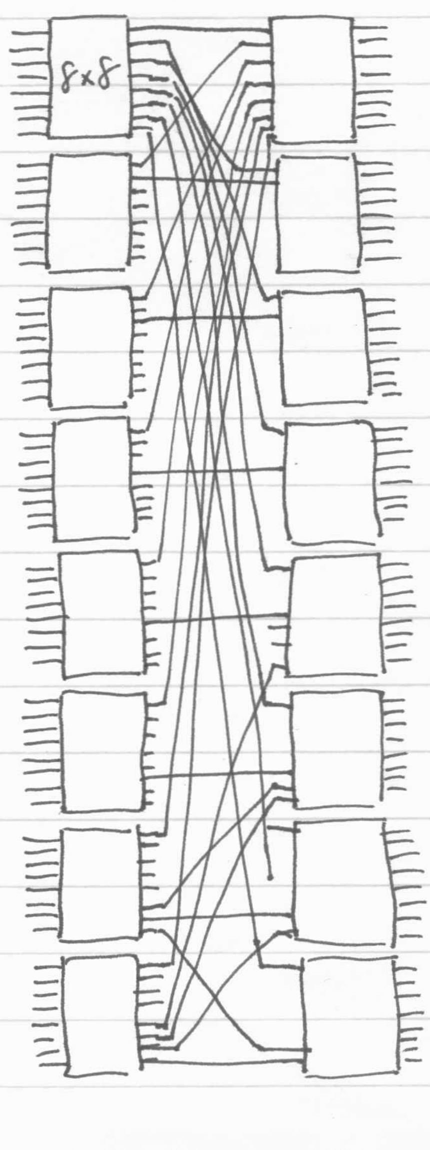
Type	Cost	Contention	Latency
 Bus	$O(n)$	Worst	1
 X BAR	$O(n^2)$	Best	1
 RING	$O(n)$	—	n
 Ω NETWORK	$O(n \log n)$	—	$\log n$
 TREE	$O(n)$	—	$\log n$
 HYPERCUBE	$O(n)$	—	$\log n$
 MESH	$O(n)$	—	$\sqrt{n}$

Omega Networks

In class, $n = 8$, $k = 2$



Example, $n = 64$, $k = 8$ (Note: NOT all connections are drawn.)



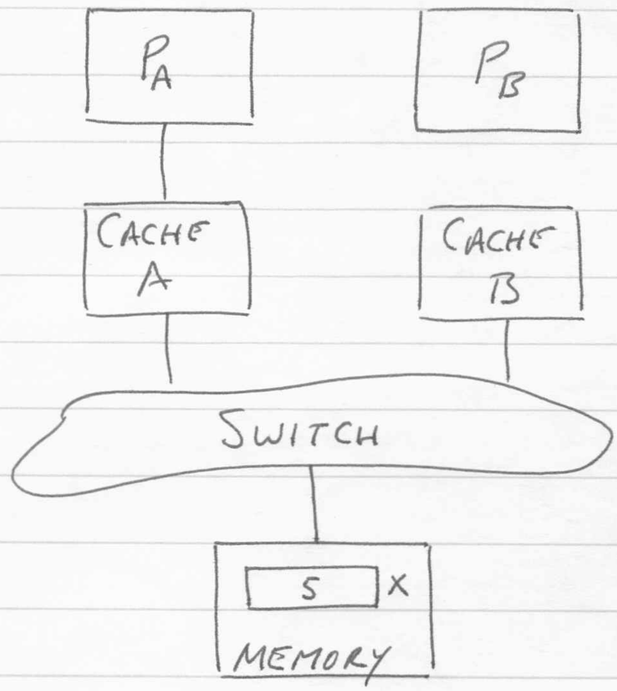
LATENCY $= \log_k n$

Cost $= k^2 \left(\frac{n}{k}\right)\left(\log_k n\right)$

CACHE COHERENCY

WHAT IS THE PROBLEM?

PROCESSORS A AND B SHARE VARIABLE X

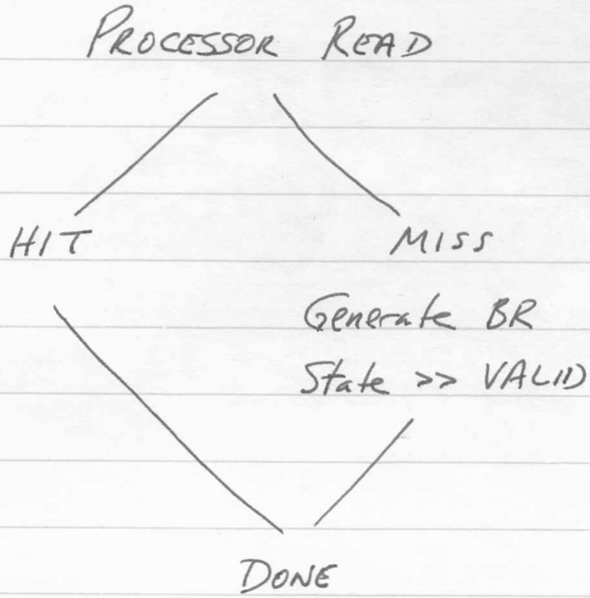


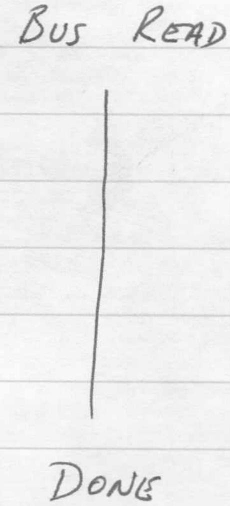
SUPPOSE THE FOLLOWING HAPPENS, IN SEQUENCE:

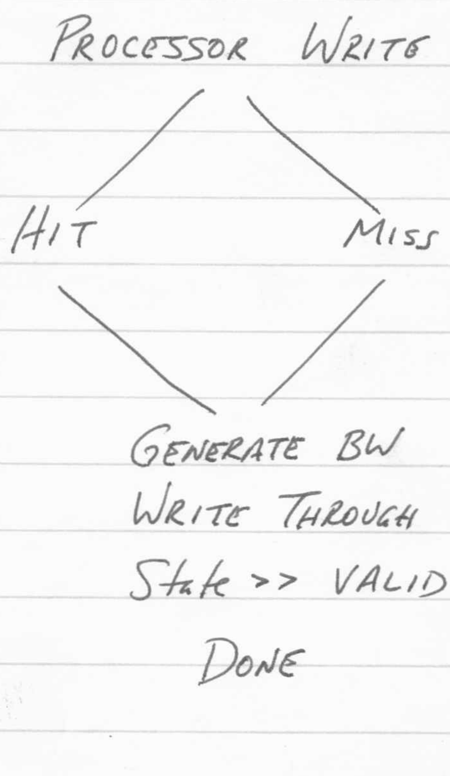
1. P_A READS X
2. P_B READS X
3. P_B WRITES THE VALUE "7" TO X
4. P_A READS X

IN STEP 4, WHAT DOES P_A READ?

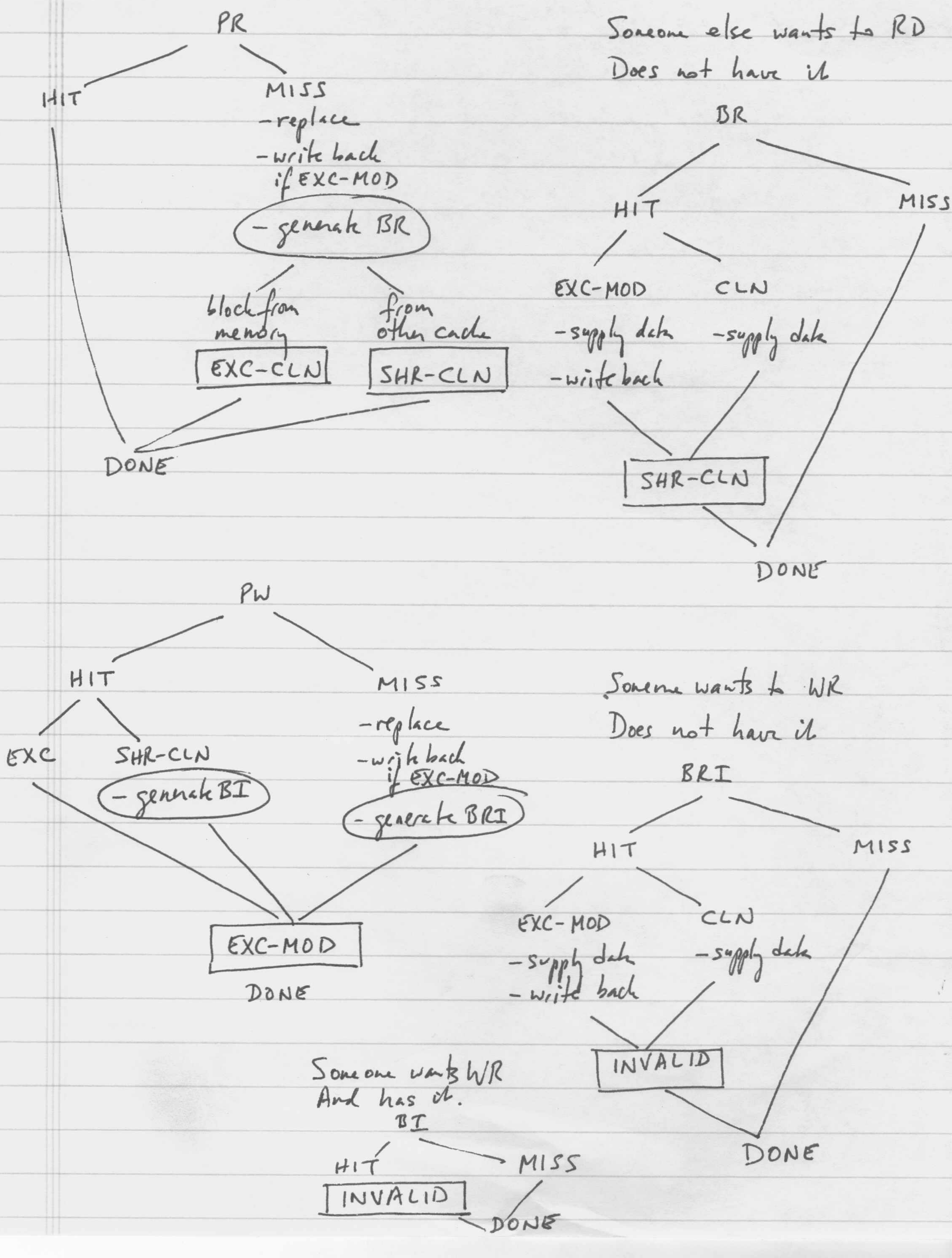
WRITE-THROUGH SCHEME







ILLINOIS


PR
HIT
MISS
-replace
-write back if EXC-MOD
- generate BR
block from memory
from other cache
EXC-CLN
SHR-CLN
DONE
Someone else wants to RD
Does not have it
BR
HIT
MISS
EXC-MOD
CLN
-supply data
-write back
-supply data
SHR-CLN
DONE
PW
HIT
MISS
EXC
SHR-CLN
- generate BI
-replace
-write back if EXC-MOD
- generate BRI
EXC-MOD
DONE
Someone wants to WR
Does not have it
BRI
HIT
MISS
EXC-MOD
CLN
-supply data
-write back
-supply data
INVALID
DONE
Someone wants WR
And has it.
BI
HIT
MISS
INVALID
DONE