Sept - 19

Note Title 19-09-2011

Homework Deadline : Thu 11:59 pm

28 - 4th (Mimr Week)

Chapter - 4

✓ Architecture { Instructions, Registers, Opcodes }

Organization { Design, Pipeline, Multiprocessor Issues }

What to Study?

MIPS Instruction Format
(Book Chapter 2)

Data Processing Instruction (32-bit RISC Format)

add, sub, and, or, nor.

MIPS → 32 registers
# of DP Instructions → 64

6 bit immediate
5 bit for shifting & other operations

Data Transfer Instructions

Very few addressing modes supported in MIPS

Register Offset (2 Register)

(2 Immediate)

Example:

```
lw  $S1, 20($S2)
```

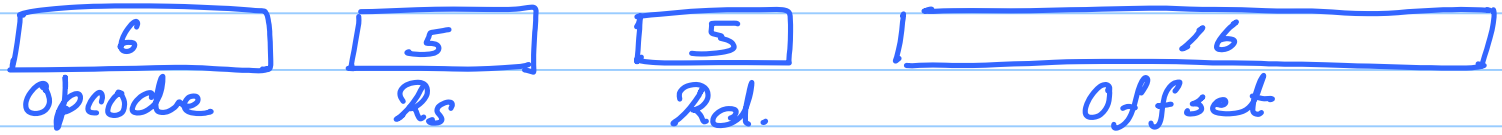
Register-Offset Addressing Mode

lw → load word

20($S2) → base register.

Address = 20 + [$S2]

Format:



lw $s1, 20 ($s2)

Rd — Offset — Source

Branch

beq $S1, $S2, 20

In MIPS, there is no CPSR

[What did we do in ARM?]

ARM (CPSR)

[CMP r1, r2
beq (20) ← some label]

cmp and branch are fused.

if ($s1 == $s2)

pc += 4 + (20×4)

+= 84

Jump / Function Call (Not there in ARM)

(0 to 2^{28} bytes)

(Very Large)

Take a look at Page -160 in the book.

Organization of the processor

(EEL 205)

List (Done Upto Now)

TTL/RTL logic families

MUX, Decoder

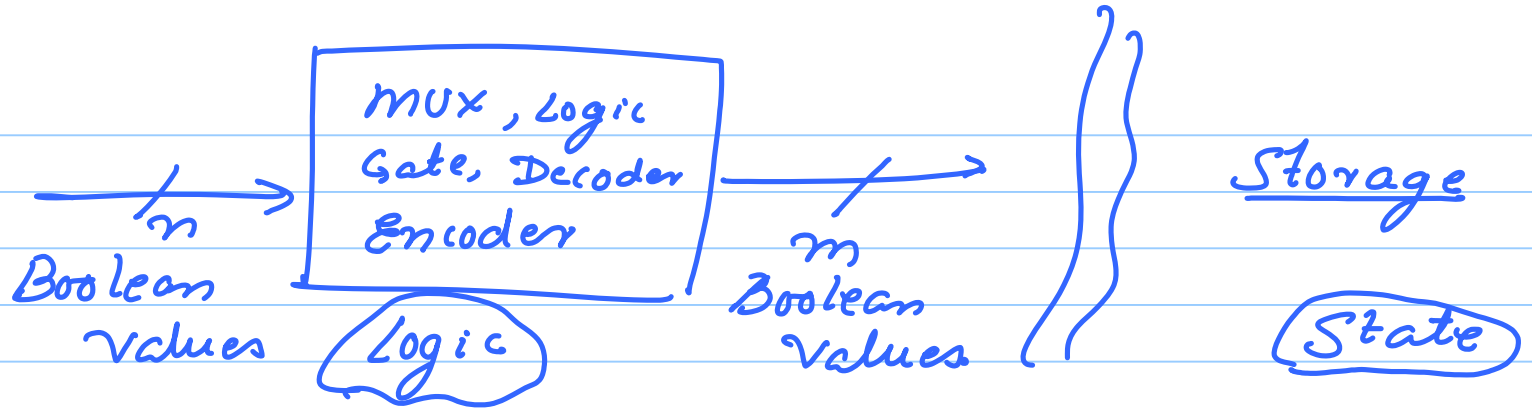
Encoder

AND/OR/XOR

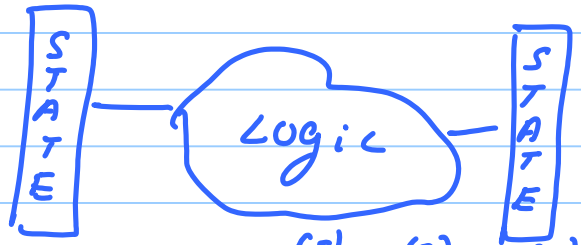
Full Adders

Minimal Amount of Background Reading. (wikipedia)

1) Latch (D - Flip Flop)
2) S-R flip flop.
3) SRAM memory cell
4) Clocking & synchronous Logic



Circuit



$(r_3 = r_1 + r_2)$

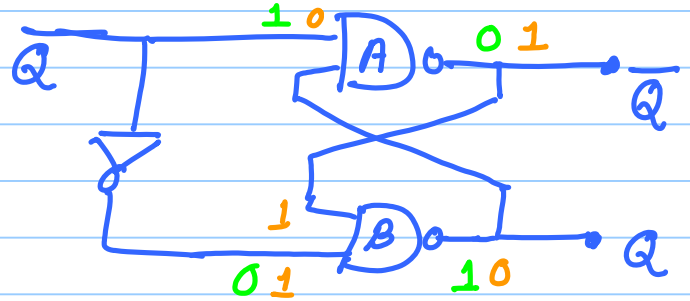
(S) 1) HW Implementations of Registers.

(L) 2) Circuitry to add two values from registers.

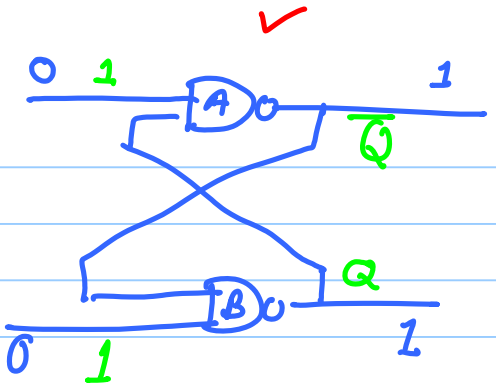
(S) 3) Register to hold the result

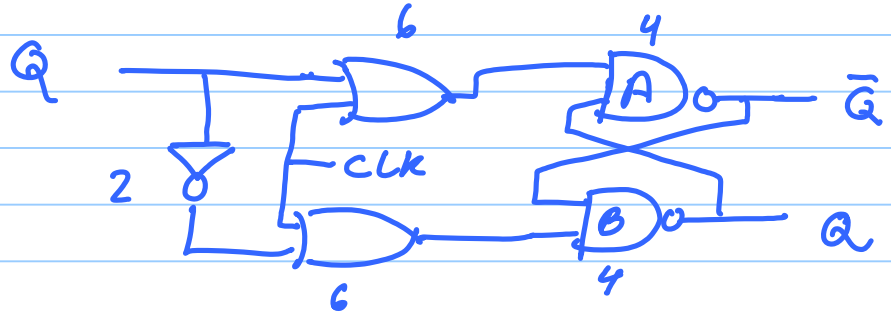
STATE Elements.

LATCH.



(Cross Coupled NAND GATES)





(22 T)

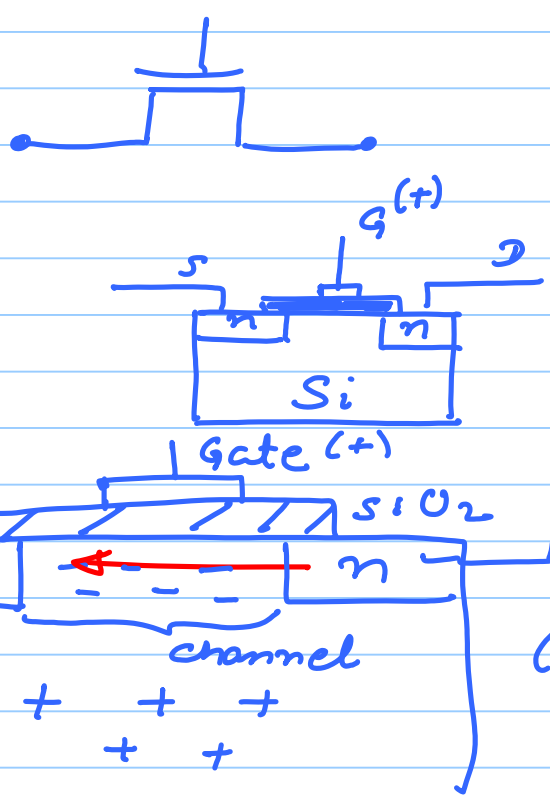
Latch

A	B	Q
0	1	0
1	0	1
0	0	(Undef)
1	1	(No change)

CLK = 1
(No Change)

CLK = 0

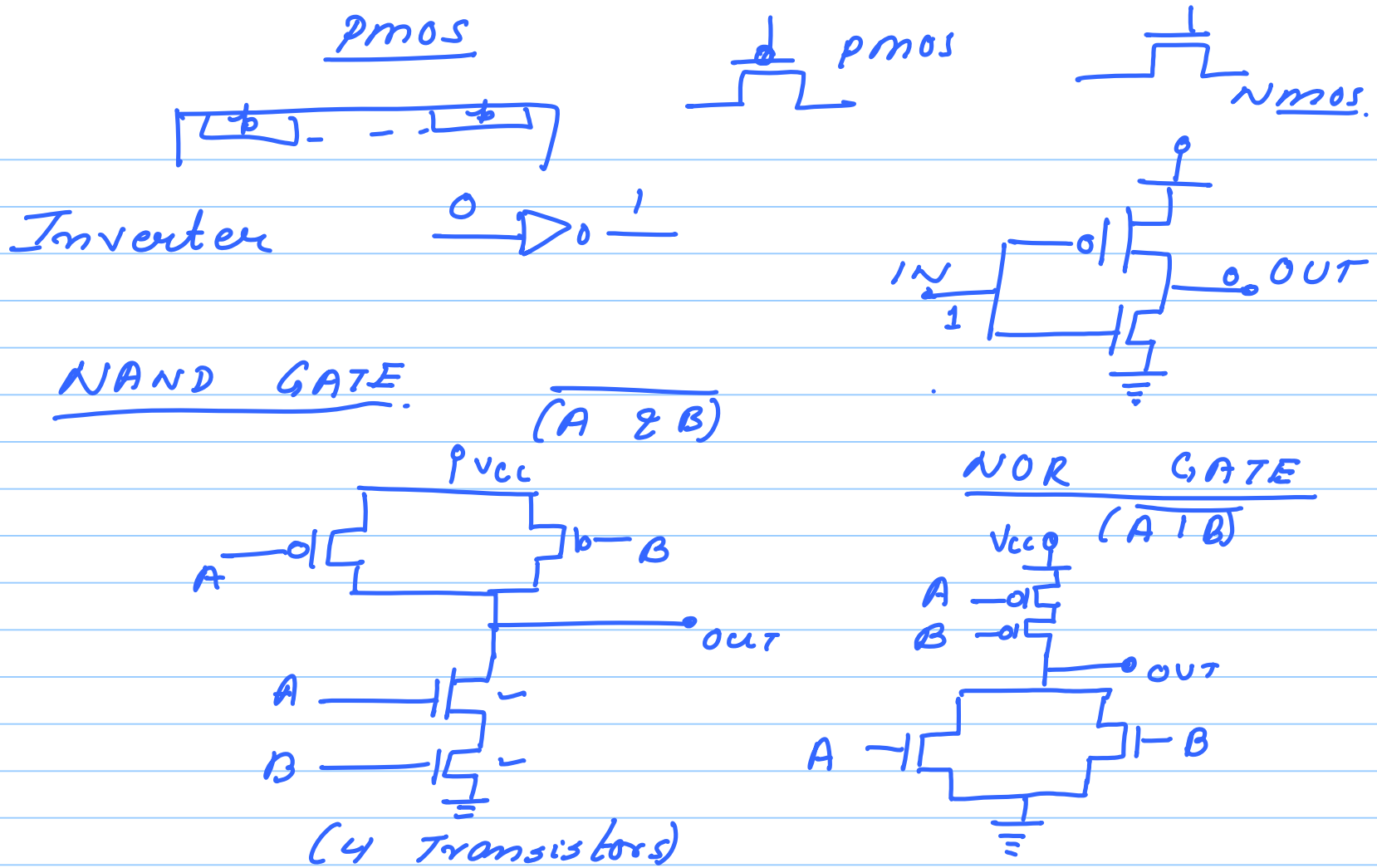
The Transistor



G → gate
S → Source
D → Drain

doping:

n → P (Electron)
p → B (Hole)


pmos
pmos
NMOS
Inverter
0
1
IN
1
OUT
NAND GATE
(A & B)
Vcc
A
B
OUT
A
B
(4 Transistors)
NOR GATE
(A | B)
Vcc
A
B
OUT
A
B

September 21st

LATCH

22 Transistors to save (1) bit

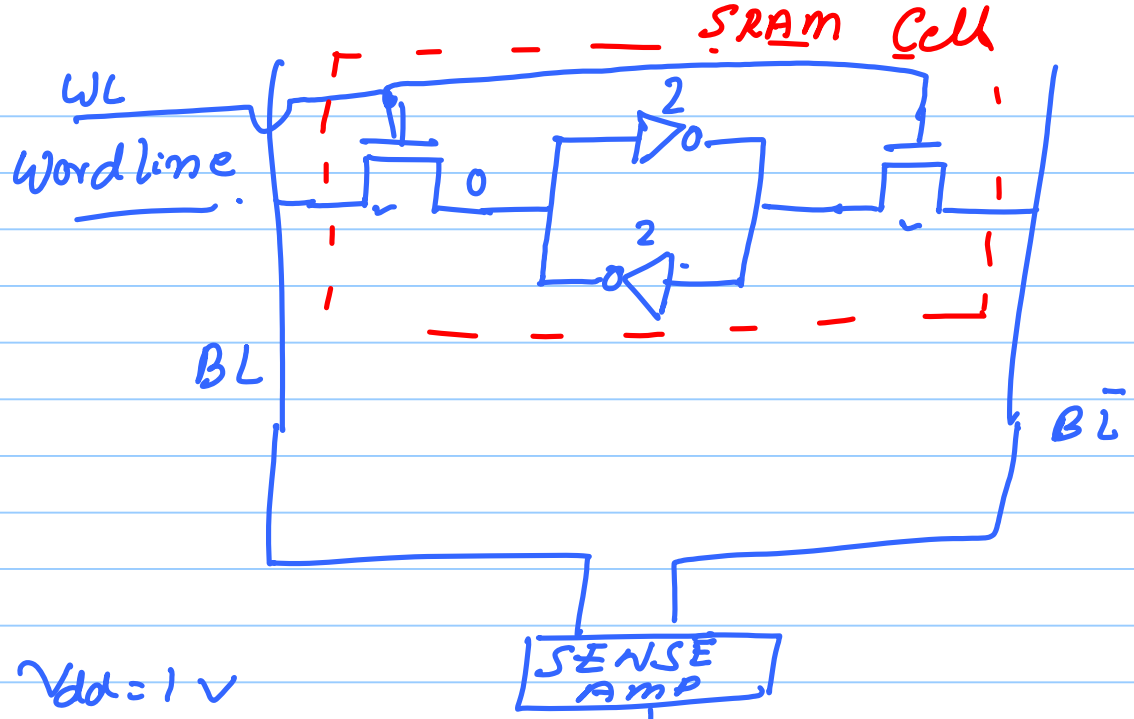
Is this a good idea !

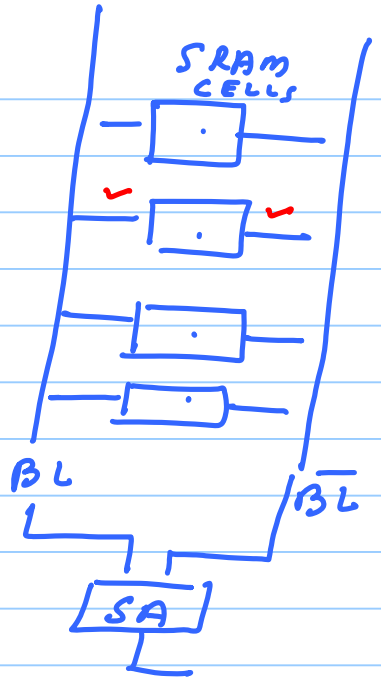
Something Better

SRAM (Static Ram)

RAM → Random Access Memory

1 bit → 6 Transistors.

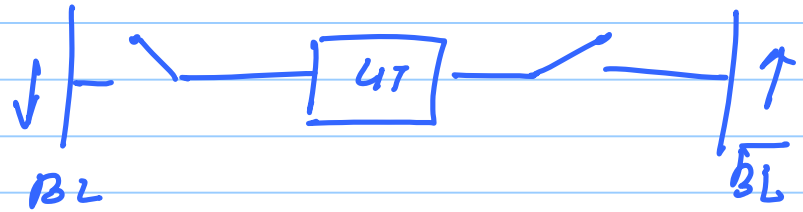




$V_{dd} = 1\,V$

1) Precharge BL & $\overline{BL}$ to $0.5\,V$

2) Set $WL = 1$



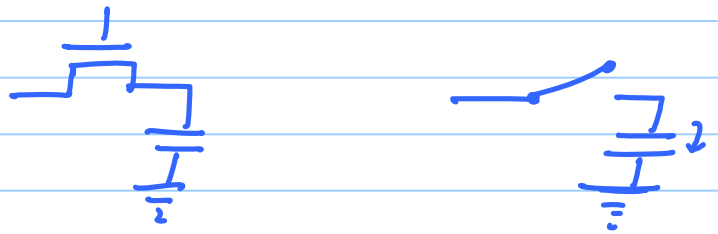
3) Monitor the voltage difference (d) in the SENSE AMP

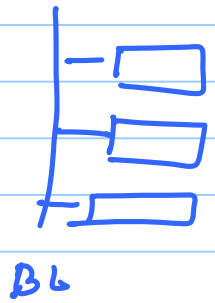
4) If $d > 100\,mV$ (result to be 1)

$d < -100\,mV$ (result is 0)

Summary :- SRAMs (6 Transistor).

Dynamic RAM (DRAM) (1 T) + (1 CAP)





Periodically Refresh → Read Value Again.
Write

	Latch	SRAM	DRAM
# Transistors	22	6	1
Area	--	ok	++
Power	--	on	++
Speed	++	+	--
Usage	Pipeline.	Caches	Main Memory

(BMW)	(HYUNDAI i10)	(Truck)
Size	Bytes	Kilo Bytes	Giga Bytes

Tutorial: Harsh Kumar.

Logi Sim