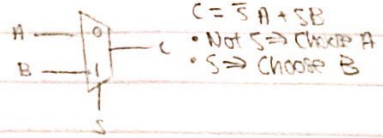
Lecture 17: Combinational Logic Blocks

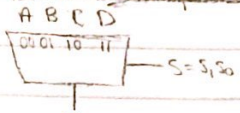
Data Multiplexer (2-to-1)

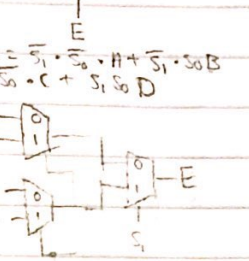


$C = \bar{S}A + SB$

- Not S ⇒ Choose A
- S ⇒ Choose B

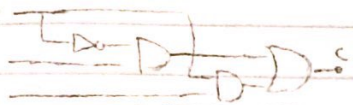
4 to 1 Multiplexer



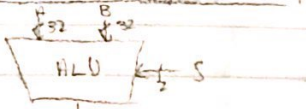


$E = \bar{S_1}\cdot\bar{S_0}\cdot A + \bar{S_1}\cdot S_0 B + S_1\cdot\bar{S_0}\cdot C + S_1 S_0 D$

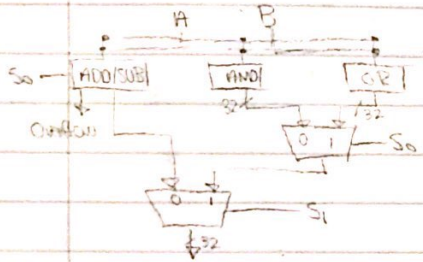
- Use 2-to-1 multiplexers to select initial A/B & C/D
- Choose b/w these w/ S_1



Arithmetic Logic Unit



- S=00 ⇒ R = A+B
- S=01 ⇒ R = A−B
- S=10 ⇒ R = A&B
- S=11 ⇒ R = A|B



- S_0 simultaneously used to select ADD/OR along w/ add/sub
- Need to deal w/ overflow for add/sub
- AND/OR just constituted by 32 series of bitwise operations

1 Bit Adder

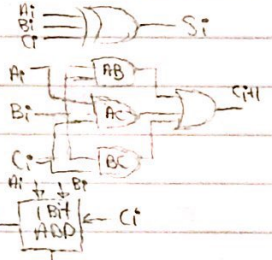
$A_i A_0$
$+ B_i B_0$
$\cdots S_i S_0$

$S_0 = a_0$ XOR b_0

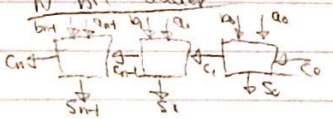
$C_1 = a_0$ AND b_0

$S_i = \text{XOR}(a_i, b_i, C_i)$

$C_{i+1} = \text{MAJ}(a_i, b_i, C_i) = a_i b_i + a_i C_i + b_i C_i$



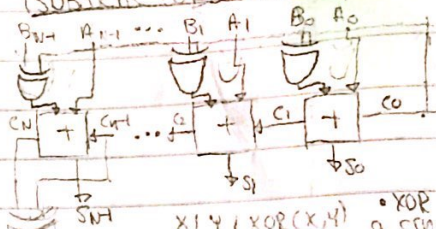
N bit adder



Overflow

- Unsigned: if $C_{n+1} = 1$ ⇒ OVERFLOW bit not enough bits to represent
- 2's C: 00 = 0, 01 = 1, 10 = −2, 11 = −1
  - 00 + Anything = Anything ⇒ Fine
  - ★ 01 + 01 (1+1) = 10 (−2) ⇒ Overflow [$C_{in}=1$, $C_{out}=0$]
  - 01 (1) + 10 (−2) = 11 (−1)
  - 01 (1) + 11 (−1) = 100 (0) ⇒ Fine ($C_{in}=1$, $C_{out}=1$)
  - ★ 10 (−2) + 10 (−2) = 100 (−4) ⇒ Overflow [$C_{in}=0$, $C_{out}=1$]
  - ★ 10 (−2) + 11 (−1) = 101 (−3) ⇒ Overflow [$C_{in}=0$, $C_{out}=1$]
  - 11 (−1) + 11 (−1) = 110 (−2) ⇒ Fine
- KEY FORMULA: Overflow = C_n XOR C_{n+1}

SUBTRACTOR: A − B = A + (−B)



X	Y	XOR (X, Y)
0	0	0
0	1	1
1	0	1
1	1	0

- XOR acts as a conditional inverter
- If SUB = 1 ⇒ Invert B's bits & add 1 to perform 2's C on it

Overflow represented by carry-in & carry-out of final adder