ECE154A — Discussion 00

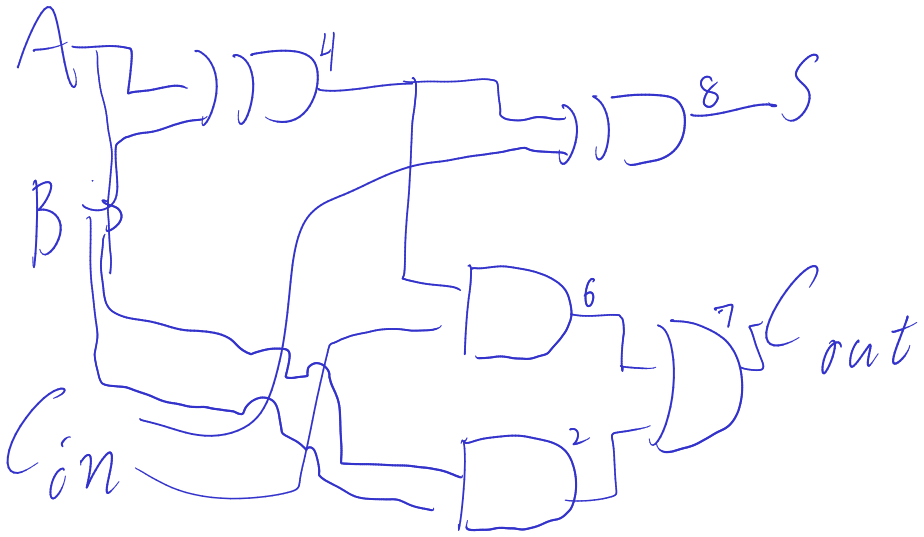
George Higgins Hutchinson

October 8, 2021

ECE154A — Discussion 00

George Higgins Hutchinson

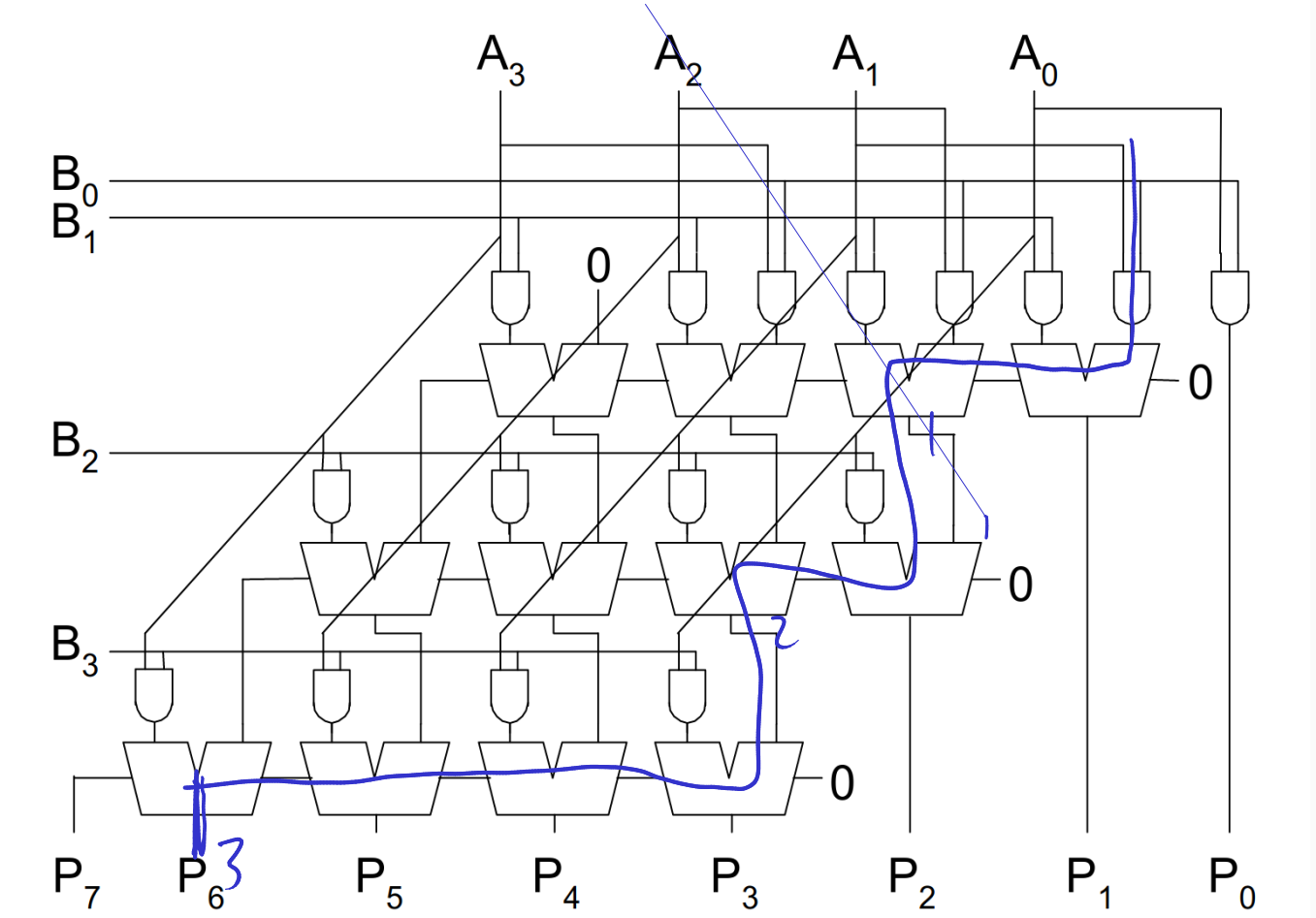
October 8, 2021


A
B
C_{in}
4
8
S
6
7
C_{out}
2

Keep your eyes open for...

- PSet 1: due ~~Wednesday, Oct 6~~ Monday, Oct 11
- Lab 2: should be officially assigned soon

Array Multiplier — Practice Problem



1 AND

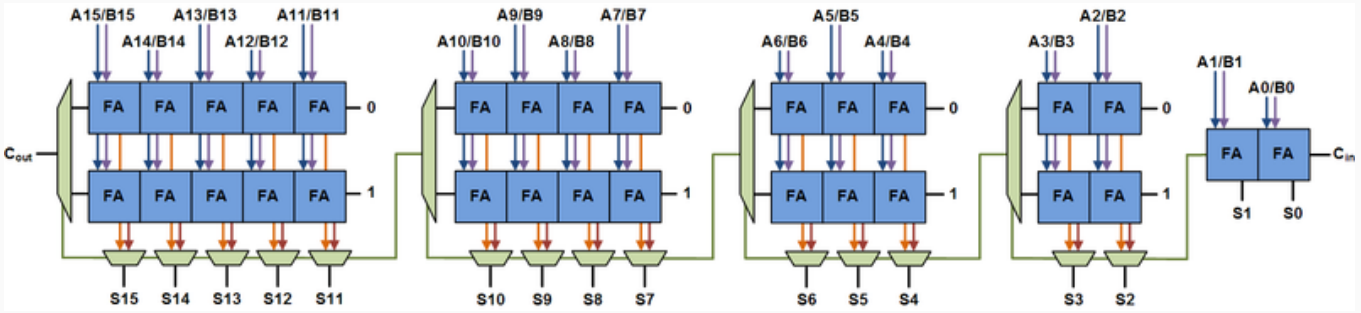
5 → Cout

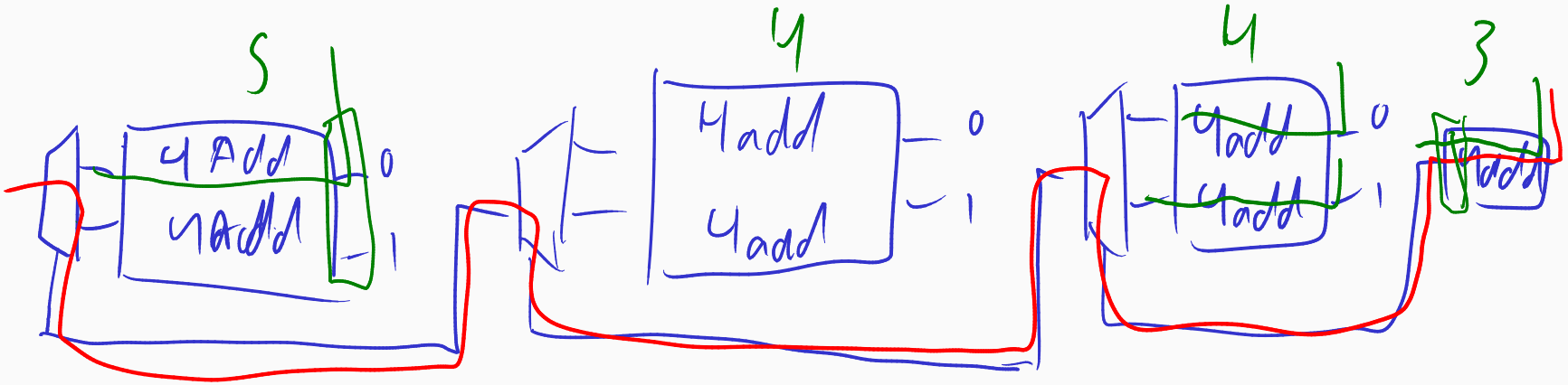
3 + 5

Let $t_{AND} = 2\text{ps}$, $t_{XOR} = 4\text{ps}$, $t_{OR} = 1\text{ps}$.

What is the critical path delay? 2ps + 5·7ps + 3·8ps = 61ps

Carry-Select Adder

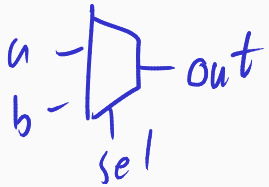




Carry-Select Adder

- Write a logic expression for the 2:1 muxes.

$out = a \cdot sel + \overline{sel} \cdot b$



- Write boolean expressions for both outputs of the full adder cells.

$S = A \oplus B \oplus C_{in}$

$C_{out} = AB + C_{in}(A \oplus B) = AB + C_{in}A + C_{in}B$

- Identify the critical path, and calculate the delay (assume path-invariant full adder)

- Derive general expressions for area and delay in an N-bit carry-select adder, split into k blocks.

Carry-Select Adder

- Write a logic expression for the 2:1 muxes.
  $Out = !Sel * a + Sel * b$
- Write boolean expressions for both outputs of the full adder cells.

- Identify the critical path, and calculate the delay (assume path-invariant full adder)

- Derive general expressions for area and delay in an N-bit carry-select adder, split into k blocks.

Carry-Select Adder

- Write a logic expression for the 2:1 muxes.
  $Out = !Sel * a + Sel * b$
- Write boolean expressions for both outputs of the full adder cells.
  $S = X \oplus B \oplus C_{in}$; $C_{out} = A * B + A * C_{in} + B * C_{in}$
- Identify the critical path, and calculate the delay (assume path-invariant full adder)

- Derive general expressions for area and delay in an N-bit carry-select adder, split into k blocks.

Carry-Select Adder

- Write a logic expression for the 2:1 muxes.
  $Out = !Sel * a + Sel * b$
- Write boolean expressions for both outputs of the full adder cells.
  $S = X \oplus B \oplus C_{in}$; $C_{out} = A * B + A * C_{in} + B * C_{in}$
- Identify the critical path, and calculate the delay (assume path-invariant full adder)
  $t = 4t_{FA} + 3t_{MUX}$
- Derive general expressions for area and delay in an N-bit carry-select adder, split into k blocks.

N
k
t_{FA}, A_{FA}
t_{MUX}, A_{MUX}

$A = (2N - k) A_{FA} + (k-1) A_{MUX}$

$t = 2k\, t_{FA} + (k-1)\, t_{MUX}$

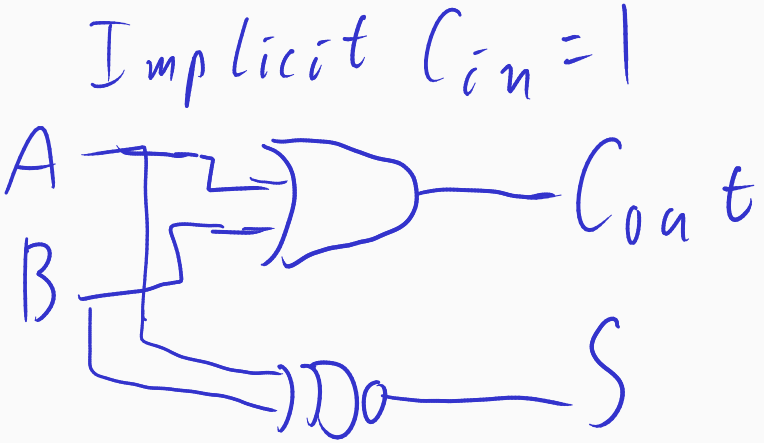
Carry-Select Adder

- Write a logic expression for the 2:1 muxes.
  $Out = !Sel * a + Sel * b$
- Write boolean expressions for both outputs of the full adder cells.
  $S = X \oplus B \oplus C_{in}$; $C_{out} = A * B + A * C_{in} + B * C_{in}$
- Identify the critical path, and calculate the delay (assume path-invariant full adder)
  $t = 4t_{FA} + 3t_{MUX}$
- Derive general expressions for area and delay in an N-bit carry-select adder, split into k blocks. in blocks of k
  $A = kA_{FA} + \frac{N}{k-1}(2kA_{FA} + (k+1)A_{MUX})$; $t = kt_{FA} + \frac{N}{k-1}t_{MUX}$

CSA, Cont...

Wait, we can replace some of those cells with half-adders (constant C_{in})...

- Write Boolean expressions and draw gate diagrams for half-adders with either constant carry in.
- Assume a delay of $\frac{1}{3}t_{FA}$ along all paths of the half-adders. Without changing the core idea, suggest a change to the structure of the adder to reduce delay.



CSA, Cont...

Wait, we can replace some of those cells with half-adders (constant C_{in})...

- Write Boolean expressions and draw gate diagrams for half-adders with either constant carry in.
- Assume a delay of $\frac{1}{3}t_{FA}$ along all paths of the half-adders. Without changing the core idea, suggest a change to the structure of the adder to reduce delay.
  Group as 5-4-4-3 instead of 4-4-4-4. Reduced initial carry chain saves a full-adder delay. Discuss: why can't we keep doing this?

Mystery Code!

```
0x00000004      main:   li $a0, 1
0x00000008              jal mystery
0x0000000c              addu $a0, $0, $v0
0x00000010              jal mystery
0x00000014              addu $a0, $0, $v0
0x00000018              jal mystery
0x0000001c              addu $a0, $0, $v0
0x00000020              jal mystery
-------------------------------------------
0x80000004      mystery: lui $t0, 0xffff
0x80000008              lui $t2, %Hi(mystery)
0x8000000c              ori $t2, %Lo(mystery)
0x80000010              addiu $t1, $0, 0
0x80000014              andi $a0, $a0, 0xffff
0x80000018              add $v0, $a0, $t1
0x8000001c              lw $t3, 12($t2)
0x80000020              and $t3, $t3, $a0
0x80000024              or $t3, $t3, $a0
0x8000002c              sw $t3, 12($t2)
0x80000030              jr $ra
```

What does it do? After 1 run? 2? N?