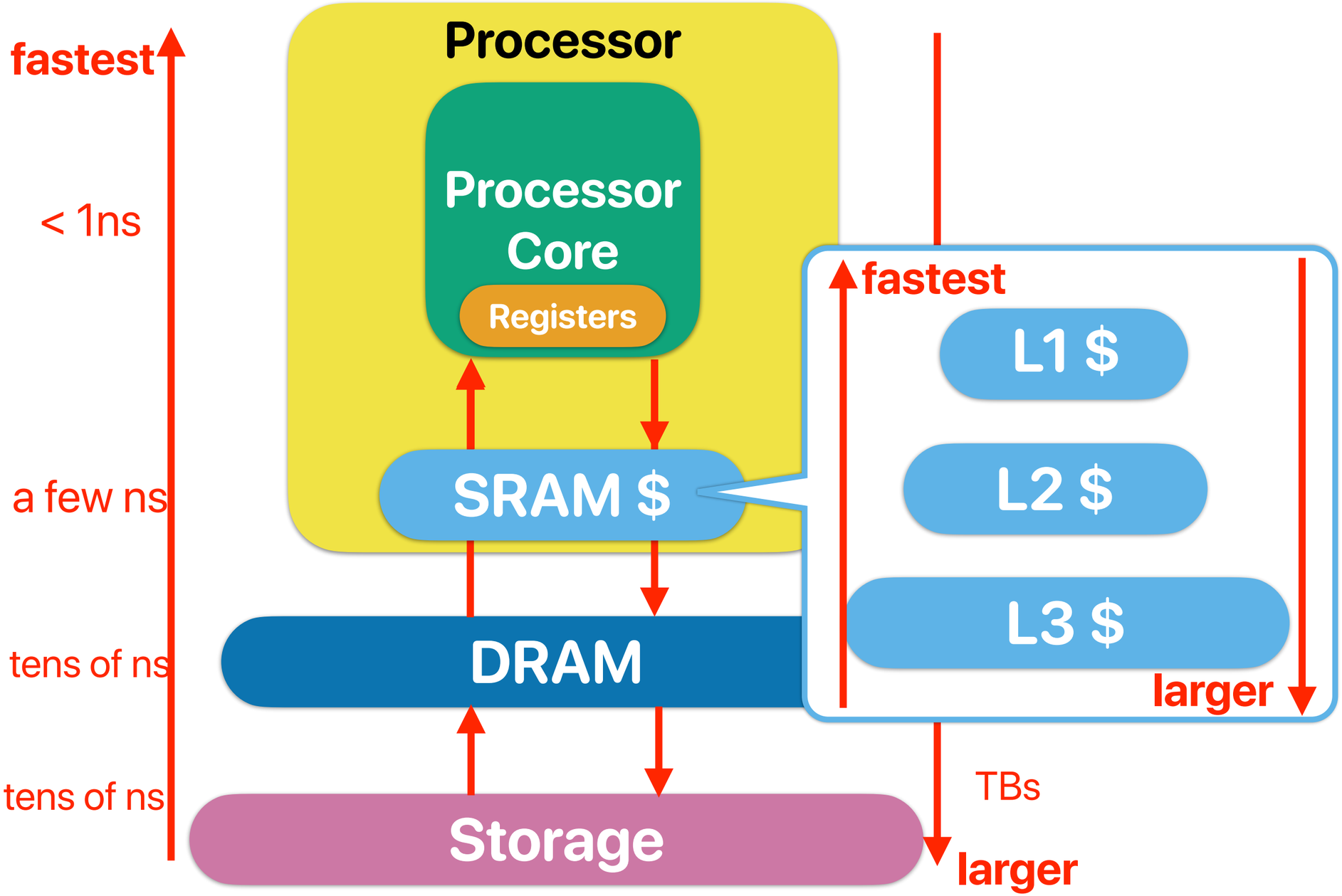
Datapath component (4)

Prof. Usagi

Recap: Memory "hierarchy" in modern processor architectures



QLC = More Density Per NAND Cell

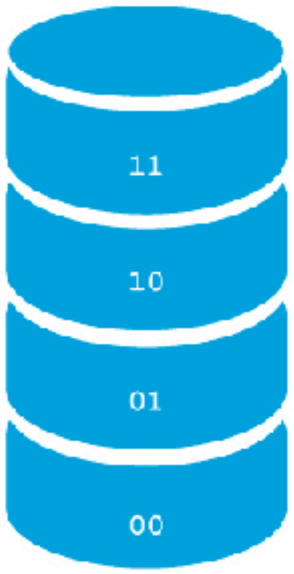
Lower $ per GB

SLC

1 Bit Per Cell

First SSD NAND technology

100K P/E Cycles
(at technology introduction)

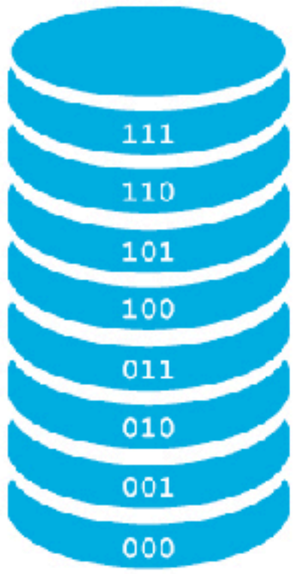


MLC

2 Bits Per Cell

100% increase

10K P/E Cycles

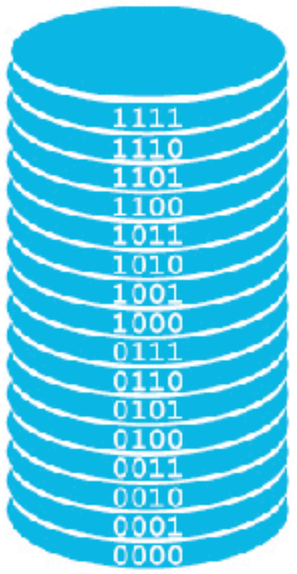


TLC

3 Bits Per Cell

50% increase

3K P/E Cycles



QLC

4 Bits Per Cell

33% increase

1K P/E Cycles

Fewer writes per cell

Micron

Recap: Flash memory characteristics

- Regarding the following flash memory characteristics, please identify how many of the following statements are correct
  - ① Flash memory cells can only be programmed with limited times
  - ② The reading latency of flash memory cells can be largely different from programming
  - ③ The latency of programming different flash memory pages can be different
  - ④ The programmed cell cannot be reprogrammed again unless its charge level is refilled to the top-level
  - A. 0
  - B. 1
  - C. 2
  - D. 3
  - E. 4

If programmer doesn't know flash "features"

- Software designer should be aware of the characteristics of underlying hardware components

Spotify is writing massive amounts of junk data to storage drives

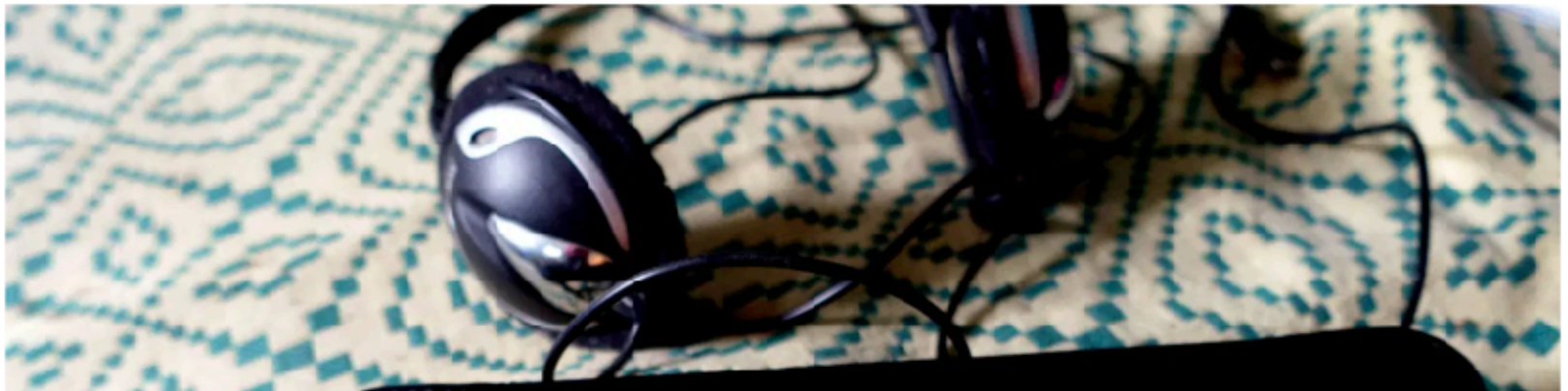
Streaming app used by 40 million writes hundreds of gigabytes per day.

DAN GOODIN - 11/10/2016, 7:00 PM

BGR TECH ENTERTAINMENT DEALS BUSINESS SCIENCE LIFESTYLE

TECH

Spotify has been quietly killing your SSD's life for months



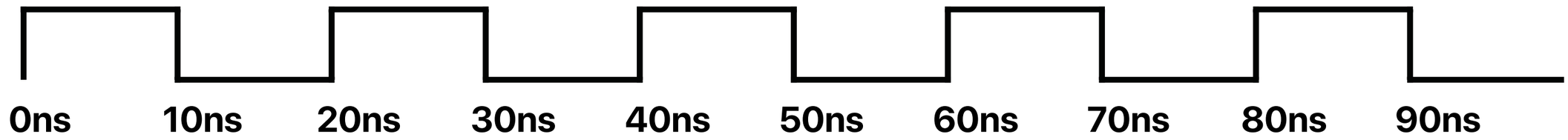
#T

1 Doctors j promisin—the Spotify music streaming app has been assaulting to potentially take years off their expected lifespans. eds of gigabytes being written in an hour aren't ed amounts are measured in terabytes. The overload

2 The mos

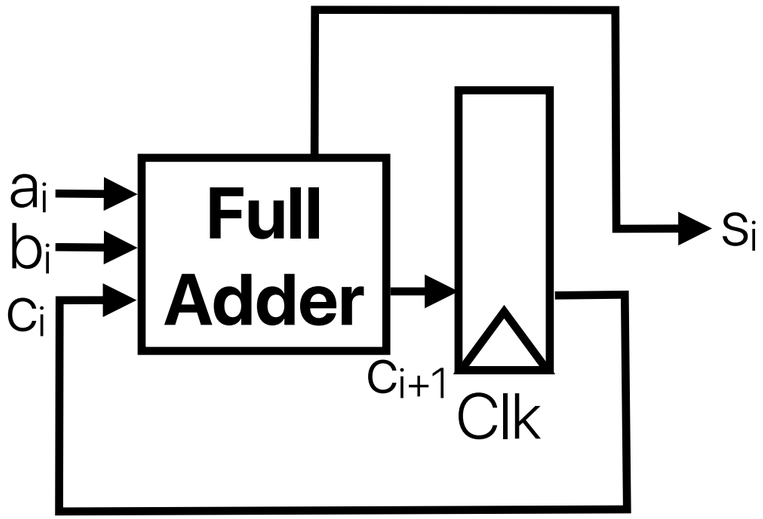
sn't storing any songs locally.

Recap: Clock signal



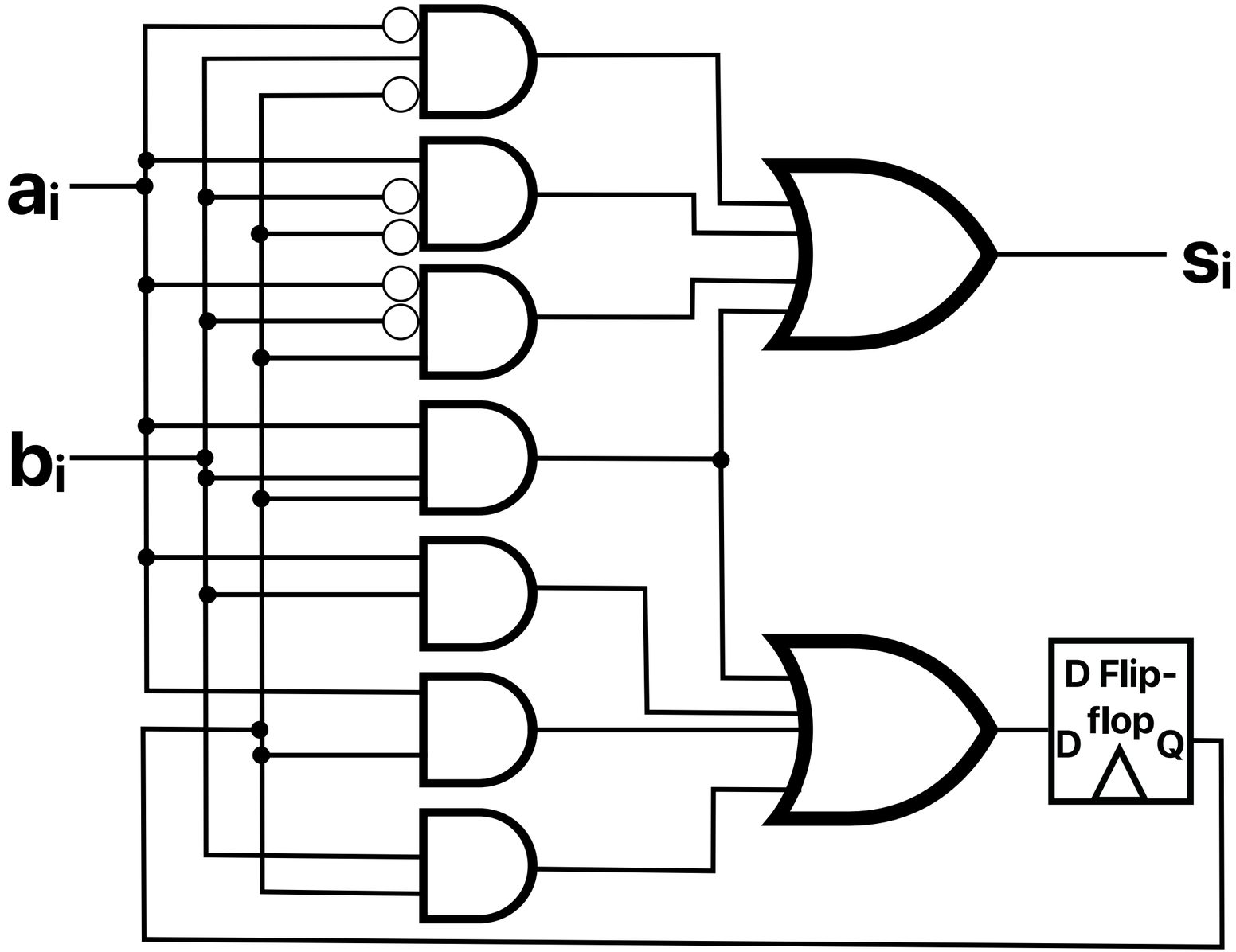
- Clock -- Pulsing signal for enabling latches; ticks like a clock
- **The clock's period must be longer than the longest delay from the state register's output to the state register's input, known as the critical path.**
- Synchronous circuit: sequential circuit with a clock
- Clock period: time between pulse starts
  - Above signal: period = 20 ns
- Clock cycle: one such time interval
  - Above signal shows 3.5 clock cycles
- Clock duty cycle: time clock is high
  - 50% in this case
- Clock frequency: 1/period
  - Above : freq = 1 / 20ns = 50MHz;

Recap: Serial Adders



Excitation Table of Serial Adder

a_i	b_i	c_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Poll close in 1:30

Critical path of the circuit?

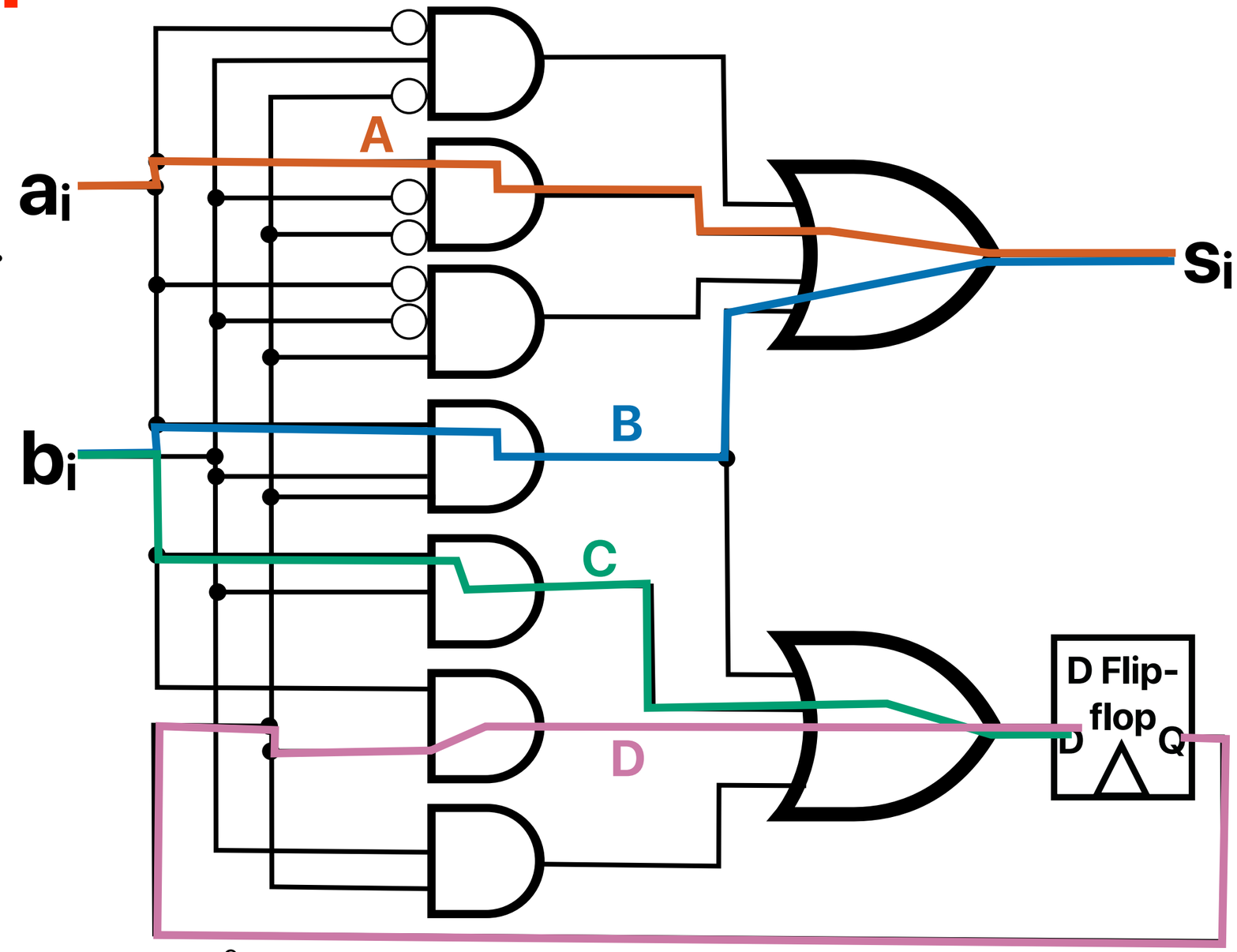
- Assume each gate delay is 1ns and the delay in a register is 2ns. Which of the following path determines the "cycle time" of the circuit?

A. A

B. B

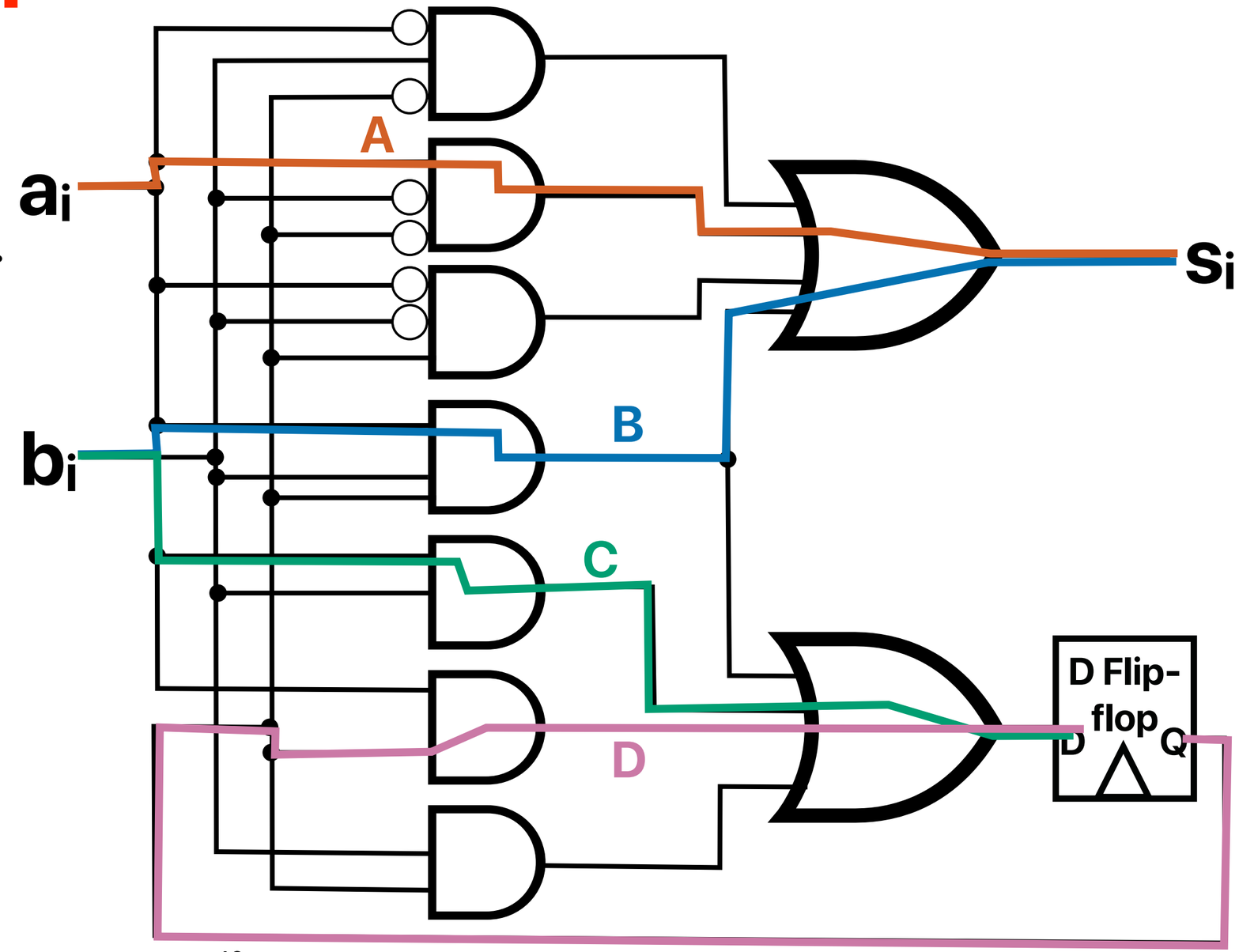
C. C

D. D



Critical path of the circuit?

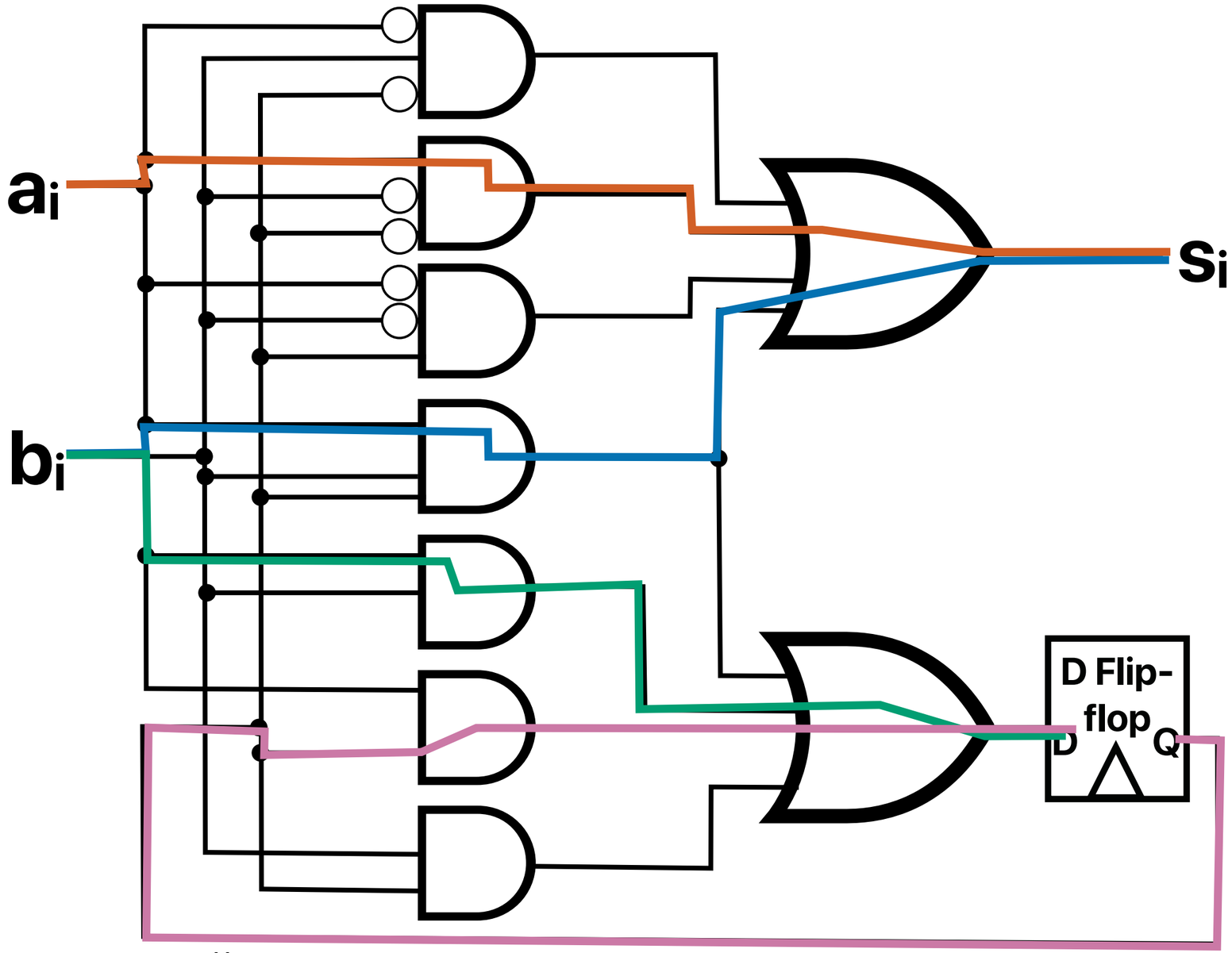
- Assume each gate delay is 1ns and the delay in a register is 2ns. Which of the following path determines the "cycle time" of the circuit?
  A. A
  B. B
  C. C
  D. D



Poll close in 1:30

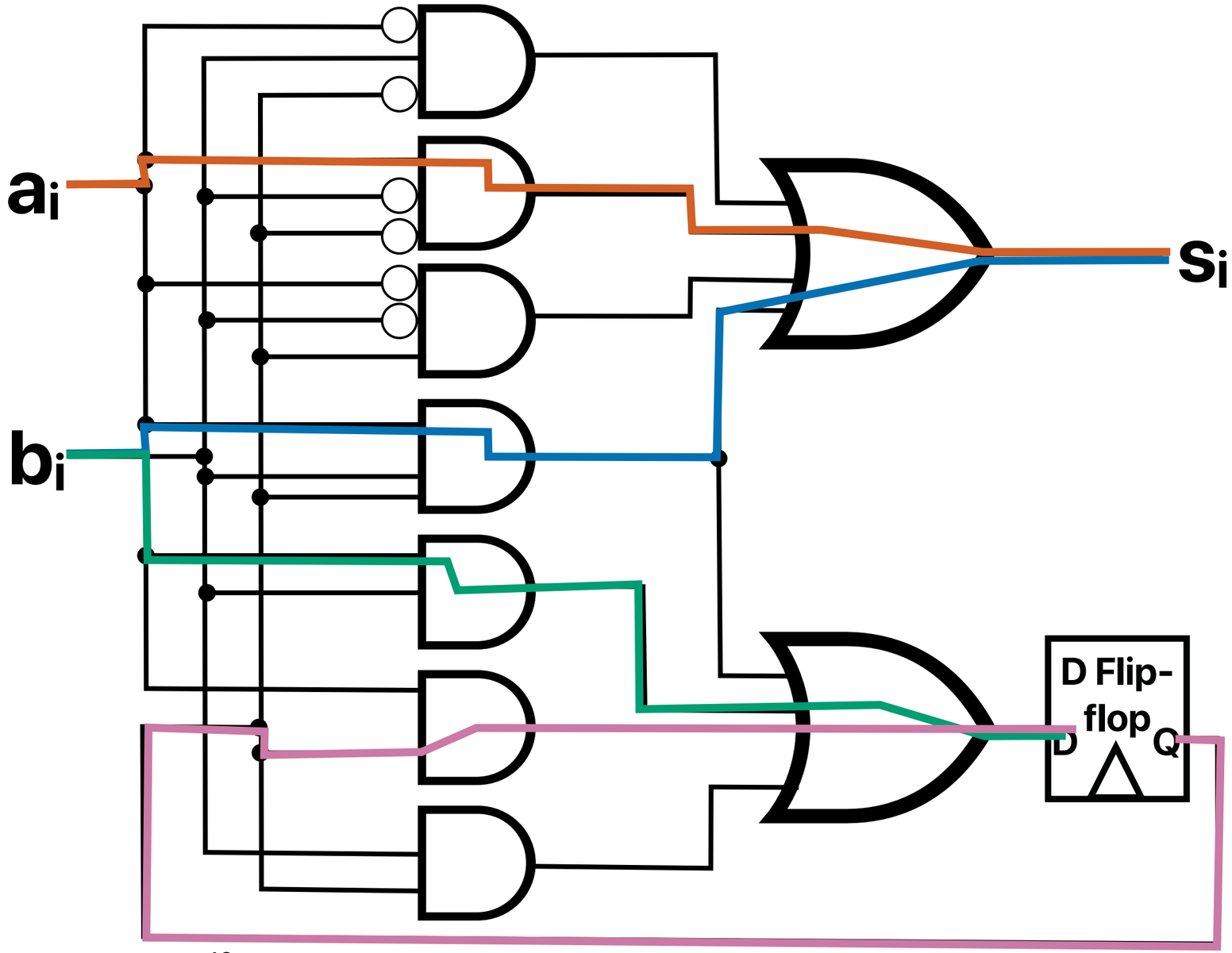
Cycle time of the circuit?

- Assume each gate delay is 1ns and the delay in a register is 2ns, what's the cycle time of the circuit?
  - A. 2 ns
  - B. 3 ns
  - C. 4 ns
  - D. 5 ns
  - E. 6 ns



Cycle time of the circuit?

- Assume each gate delay is 1ns and the delay in a register is 2ns, what's the cycle time of the circuit?
  A. 2 ns
  B. 3 ns
  C. 4 ns
  D. 5 ns
  E. 6 ns



Recap: Frequency

- Consider the following adders. Assume each gate delay is 1ns and the delay in a register is 2ns. Please rank their maximum operating frequencies
  - ① 32-bit CLA made with 8 4-bit CLA adders $\frac{1}{17ns} = 58.8MHz$
  - ② 32-bit CRA made with 32 full adders $\frac{1}{64ns} = 15.6MHz$
  - ③ 32-bit serial adders made with 4-bit CLA adders $\frac{1}{5ns} = 200MHz$
  - ④ 32-bit serial adders made with 1-bit full adders $\frac{1}{4ns} = 250MHz$
  - A. (1) > (2) > (3) > (4)
  - B. (2) > (1) > (4) > (3)
  - C. (2) > (1) > (3) > (4)
  - D. (4) > (3) > (2) > (1)
  - E. (4) > (3) > (1) > (2)

Recap: Area/Delay of adders

- Consider the following adders?
  - ① 32-bit CLA made with 8 4-bit CLA adders
  - ② 32-bit CRA made with 32 full adders
  - ③ 32-bit serial adders made with 4-bit CLA adders
  - ④ 32-bit serial adders made with 1-bit full adders

Each CLA — 2-gate delay — 8*2+1 ~ 17

Each carry — 2-gate delay — 64

Each CLA — (3-gate delay + 2-gate delay)*8 cycles — 5*8+1 = 41

Each CLA — (2-gate delay + 2-gate delay)*32 cycles — 4*32 = 128

A. Area: (1) > (2) > (3) > (4) Delay: (1) < (2) < (3) < (4)

B. Area: (1) > (3) > (2) > (4) Delay: (1) < (3) < (2) < (4)

C. Area: (1) > (3) > (4) > (2) Delay: (1) < (3) < (4) < (2)

D. Area: (1) > (2) > (3) > (4) Delay: (1) < (3) < (2) < (4)

E. Area: (1) > (3) > (2) > (4) Delay: (1) < (3) < (4) < (2)

Frequency != End-to-end latency

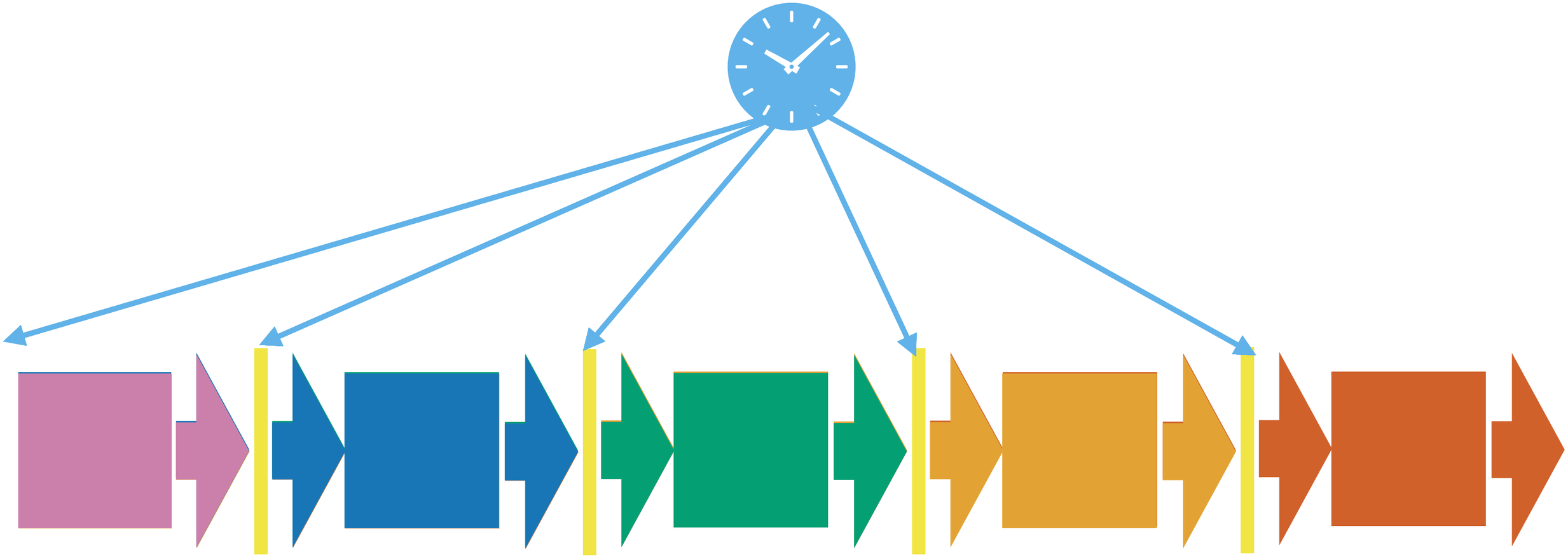
Outline

- Pipelining
- Multipliers

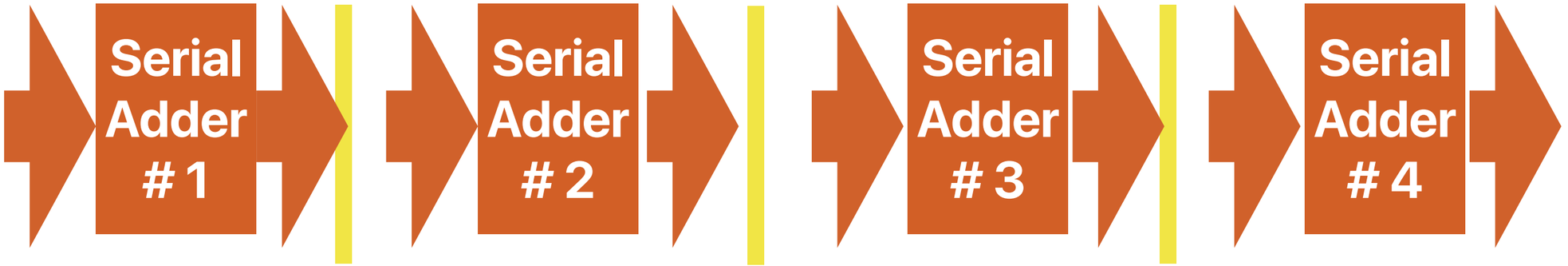
Pipelining

- Different parts of the hardware works on different requests/ commands simultaneously
- A clock signal controls and synchronize the beginning and the end of each part/**stage** of the work
- A **pipeline register** between different parts of the hardware to keep intermediate results necessary for the upcoming work
  - Register is basically an array of flip-flops!

Pipelining

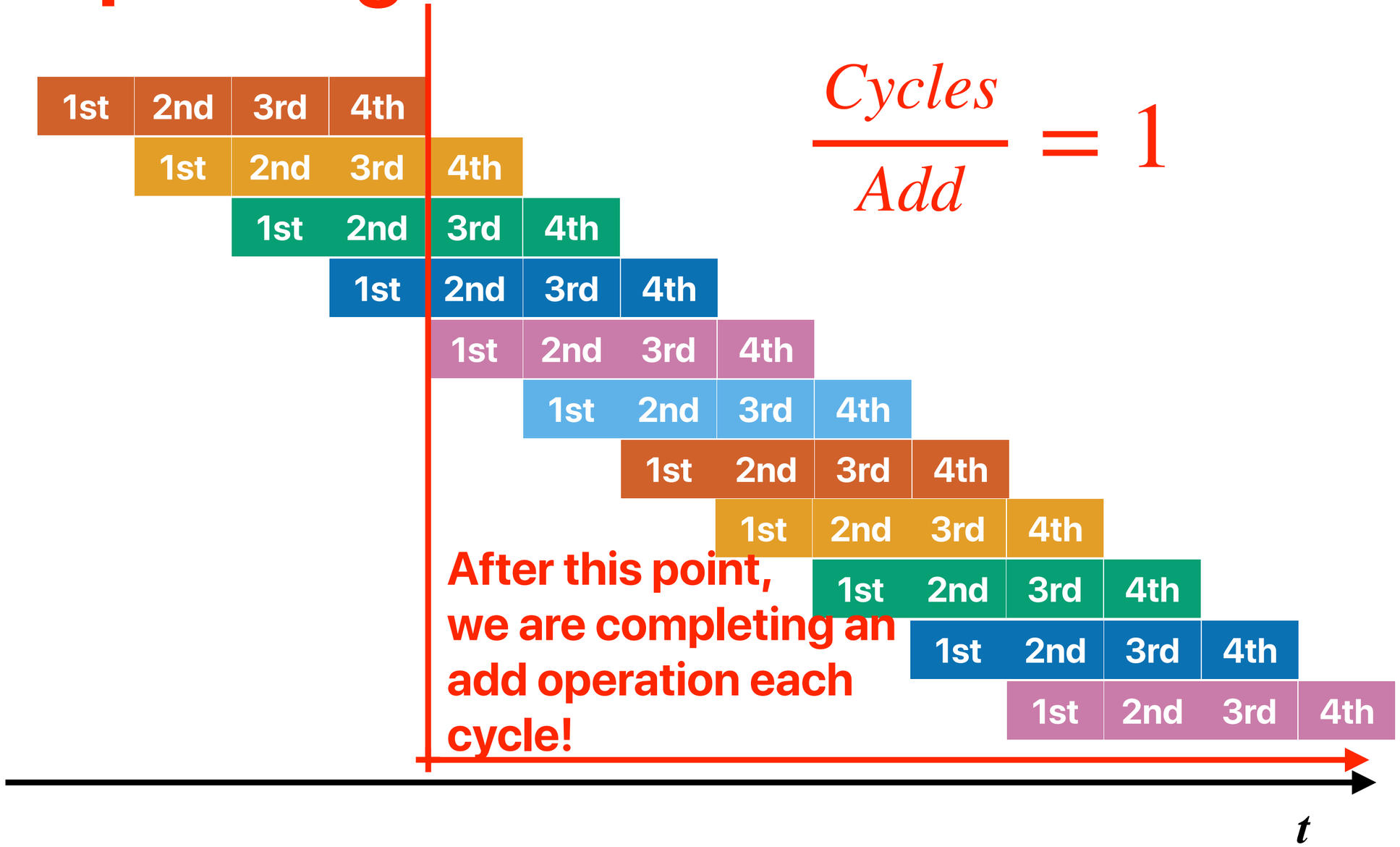


Pipelining a 4-bit serial adder



Pipelining a 4-bit serial adder

```
add a, b
add c, d
add e, f
add g, h
add i, j
add k, l
add m, n
add o, p
add q, r
add s, t
add u, v
```



$$\frac{Cycles}{Add} = 1$$

Poll close in 1:30

What if we have millions of adds to do?

- Consider the following adders. Assume each gate delay is 1ns and the delay in a register is 2ns. And we are processing 10 million of add operations. Please rank their total time in finishing these 10 million adds.
  - ① 32-bit CLA made with 8 4-bit CLA adders
  - ② 32-bit CRA made with 32 full adders
  - ③ 8-stage, pipelined 32-bit serial adders made with 4-bit CLA adders
  - ④ 32-stage, pipelined 32-bit serial adders made with 1-bit full adders
  - A. (1) < (2) < (3) < (4)
  - B. (2) < (1) < (4) < (3)
  - C. (3) < (4) < (2) < (1)
  - D. (4) < (3) < (2) < (1)
  - E. (4) < (3) < (1) < (2)

What if we have millions of adds to do?

- Consider the following adders. Assume each gate delay is 1ns and the delay in a register is 2ns. And we are processing 10 million of add operations. Please rank their total time in finishing these 10 million adds.
  - ① 32-bit CLA made with 8 4-bit CLA adders
  - ② 32-bit CRA made with 32 full adders
  - ③ 8-stage, pipelined 32-bit serial adders made with 4-bit CLA adders
  - ④ 32-stage, pipelined 32-bit serial adders made with 1-bit full adders
  - A. (1) < (2) < (3) < (4)
  - B. (2) < (1) < (4) < (3)
  - C. (3) < (4) < (2) < (1)
  - D. (4) < (3) < (2) < (1)
  - E. (4) < (3) < (1) < (2)

Latency/Delay v.s. Bandwidth/Throughput

- Latency — the amount of time to finish an operation
  - access time
  - response time
- Throughput — the amount of work can be done within a given period of time
  - bandwidth (MB/Sec, GB/Sec, Mbps, Gbps)
  - IOPs
  - MFLOPs

Latency/Delay v.s. Throughput

	Toyota Prius	100 Gb Network
	•100 miles (161 km) from UCSD •75 MPH on highway! •Max load: 374 kg = 2,770 hard drives (2TB per drive)	•100 miles (161 km) from UCSD •Lightspeed! — $3*10^8$m/sec •Max load:4 lanes operating at 25GHz
bandwidth	290GB/sec	100 Gb/s or 12.5GB/sec
latency	3.5 hours	2 Peta-byte over 167772 seconds = 1.94 Days
response time	You see nothing in the first 3.5 hours	You can start watching the movie as soon as you get a frame!

Poll close in 1:30

Area/Cost

- Consider the following adders. Please rank the number of transistors in implementing each of them
  - ① 32-bit CLA made with 8 4-bit CLA adders
  - ② 32-bit CRA made with 32 full adders
  - ③ 8-stage, pipelined 32-bit serial adders made with 4-bit CLA adders
  - ④ 32-stage, pipelined 32-bit serial adders made with 1-bit full adders

A. (1) > (2) > (3) > (4)
B. (2) > (1) > (4) > (3)
C. (3) > (4) > (2) > (1)
D. (4) > (3) > (2) > (1)
E. (4) > (3) > (1) > (2)

Recap: CLA's size

- How many transistors do we need to implement a 4-bit CLA logic?
  - A. 38
  - B. 64
  - C. 88
  - D. 116
  - E. 128

$S_i = A_i \text{ XOR } B_i \text{ XOR } C_i$

$G_i = A_i B_i$

$P_i = A_i \text{ XOR } B_i$

$C_1 = G_0 + P_0 C_0$ 4 + 4 = 8

$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0)$

$= G_1 + P_1 G_0 + P_1 P_0 C_0$

4 + 6 + 6 = 16

$C_3 = G_2 + P_2 C_2$

$= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$

$C_4 = G_3 + P_3 C_3$ 4 + 6 + 8 + 8 =26

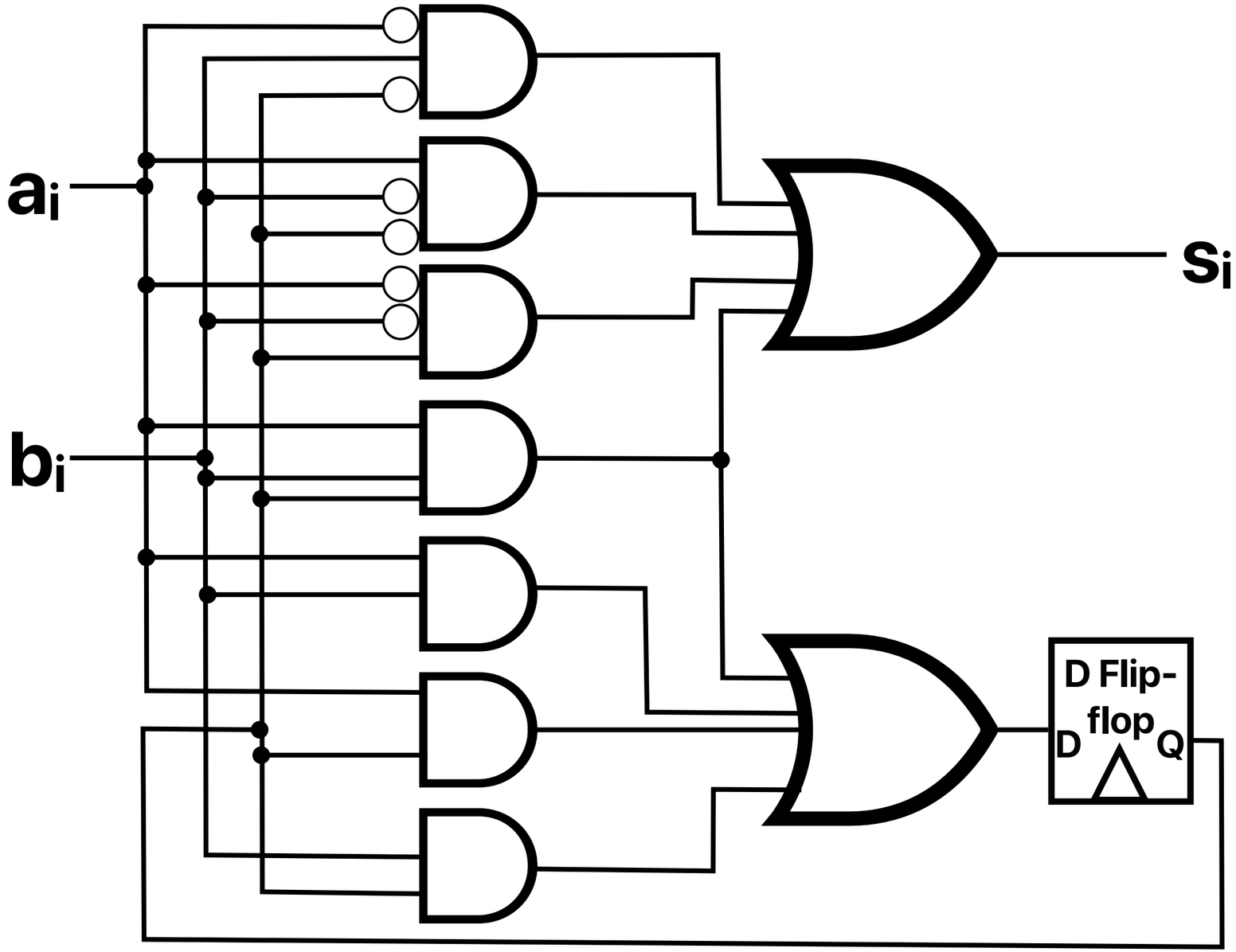
$= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0$

$+ P_3 P_2 P_1 P_0 C_0$

4 + 6 + 8 + 10 + 10 = 38

Recap: Excitation Table of Serial Adder

a_i	b_i	c_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Area/Cost

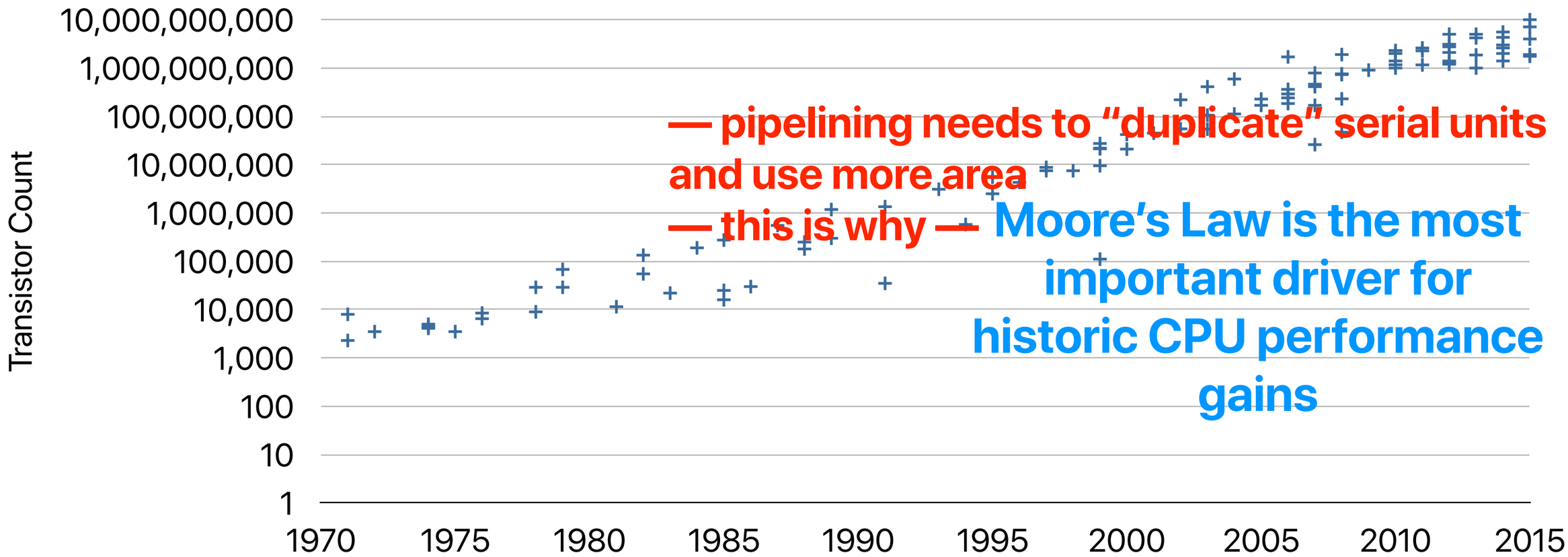
- Consider the following adders. Please rank the number of transistors in implementing each of them
  - ① 32-bit CLA made with 8 4-bit CLA adders — 1952 transistors
  - ② 32-bit CRA made with 32 full adders — 1600 transistors
  - ③ 8-stage, pipelined 32-bit serial adders made with 4-bit CLA adders — (244 transistors)*8 + 7+ (8+12+16+20+24+28+32)*18 transistors= 4479
  - ④ 32-stage, pipelined 32-bit serial adders made with 1-bit full adders — (50 transistors)*32 + (2+…+32)*18 transistors= 2127

  A. (1) > (2) > (3) > (4)
  B. (2) > (1) > (4) > (3)
  C. (3) > (4) > (2) > (1)
  D. (4) > (3) > (2) > (1)
  E. (4) > (3) > (1) > (2)

— pipelining needs to "duplicate" serial units and use more area

Moore's Law[1]

- The number of transistors we can build in a fixed area of silicon doubles every 12 ~ 24 months.



(1) Moore, G. E. (1965), 'Cramming more components onto integrated circuits', Electronics 38 (8) .

Multiplier

Binary multiplication

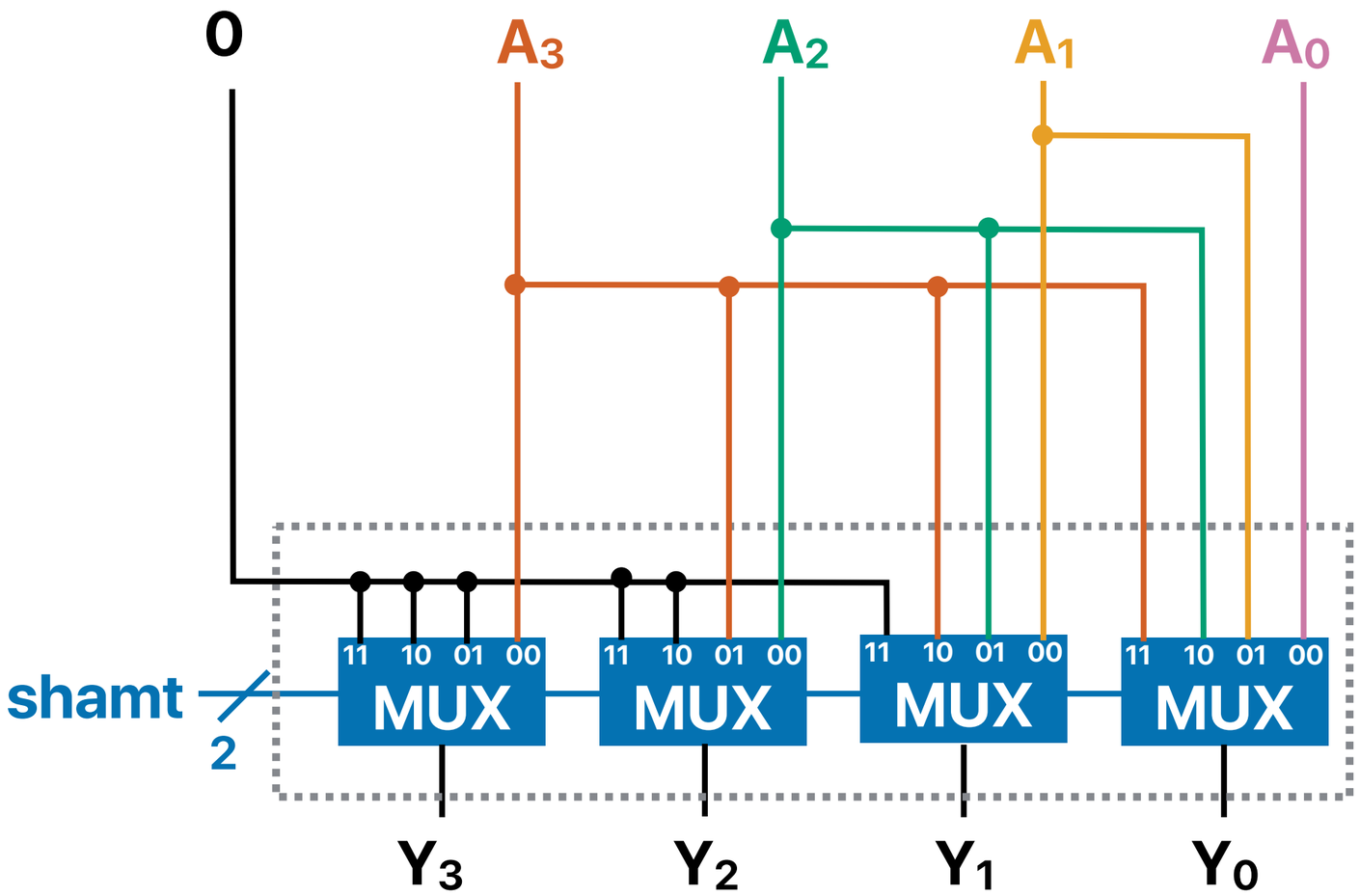
- Thinking about how you do this by hand in decimal!

```
    1 2 3 4          0 1 1 1
  × 5 6 7 8        × 1 1 0 0
-----------      -----------
    9 8 7 2          0 0 0 0
  8 6 3 8          0 0 0 0
7 4 0 4          0 1 1 1
6 1 7 0        0 1 1 1
-----------      -----------
7 0 0 6 6 5 2    1 0 1 0 1 0 0
```

			a_3	a_2	a_1	a_0	
		×	b_3	b_2	b_1	b_0	
pp1			a_3b_0	a_2b_0	a_1b_0	a_0b_0	
pp2		a_3b_1	a_2b_1	a_1b_1	a_0b_1	0	
pp3	a_3b_2	a_2b_2	a_1b_2	a_0b_2	0	0	
pp4	a_3b_3	a_2b_3	a_1b_3	a_0b_3	0	0	0
	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

Shifters

Recap: Shift "Right"



Example:
if S = 11
then
Y3 = 0
Y2 = 0
Y1 = 0
Y0 = A3

Example:
if S = 10
then
Y3 = 0
Y2 = 0
Y1 = A3
Y0 = A2

Example:
if S = 01
then
Y3 = 0
Y2 = A3
Y1 = A2
Y0 = A1

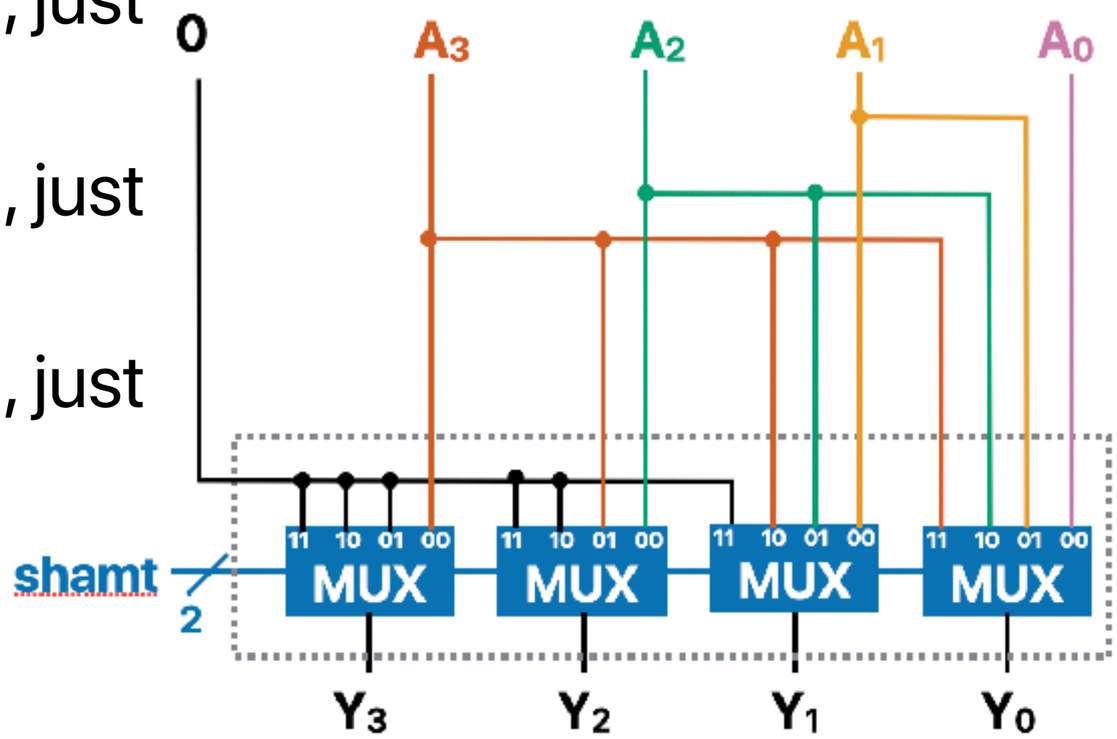
The "chain" of multiplexers determines how many bits to shift

Based on the value of the selection input (shamt = shift amount)

Poll close in 1:30

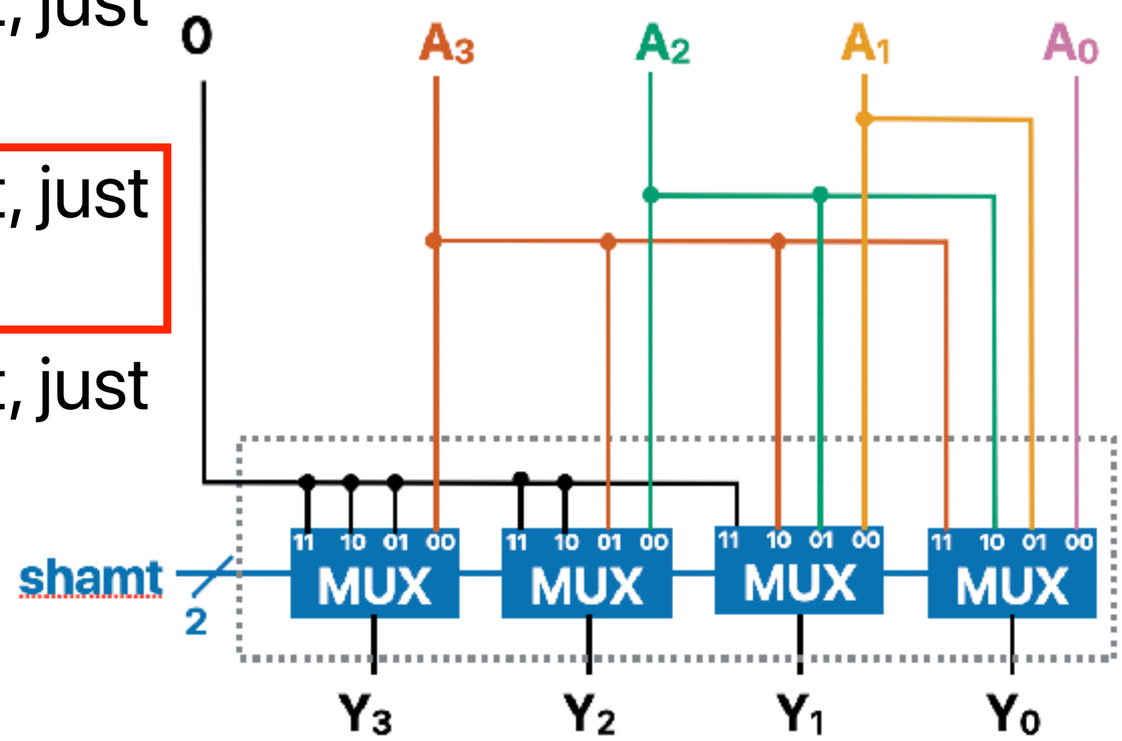
How to support shift left?

- Refer to the shift right logic, what do we need to modify to perform shift left?
  A. We can alter the interpretation of shamt to support shift left
  B. We don't need to modify the circuit, just take a not on every input
  C. We don't need to modify the circuit, just change the order of inputs
  D. We don't need to modify the circuit, just change the order of outputs
  E. None of the above



How to support shift left?

- Refer to the shift right logic, what do we need to modify to perform shift left?
  - A. We can alter the interpretation of shamt to support shift left
  - B. We don't need to modify the circuit, just take a not on every input
  - C. We don't need to modify the circuit, just change the order of inputs
  - D. We don't need to modify the circuit, just change the order of outputs
  - E. None of the above

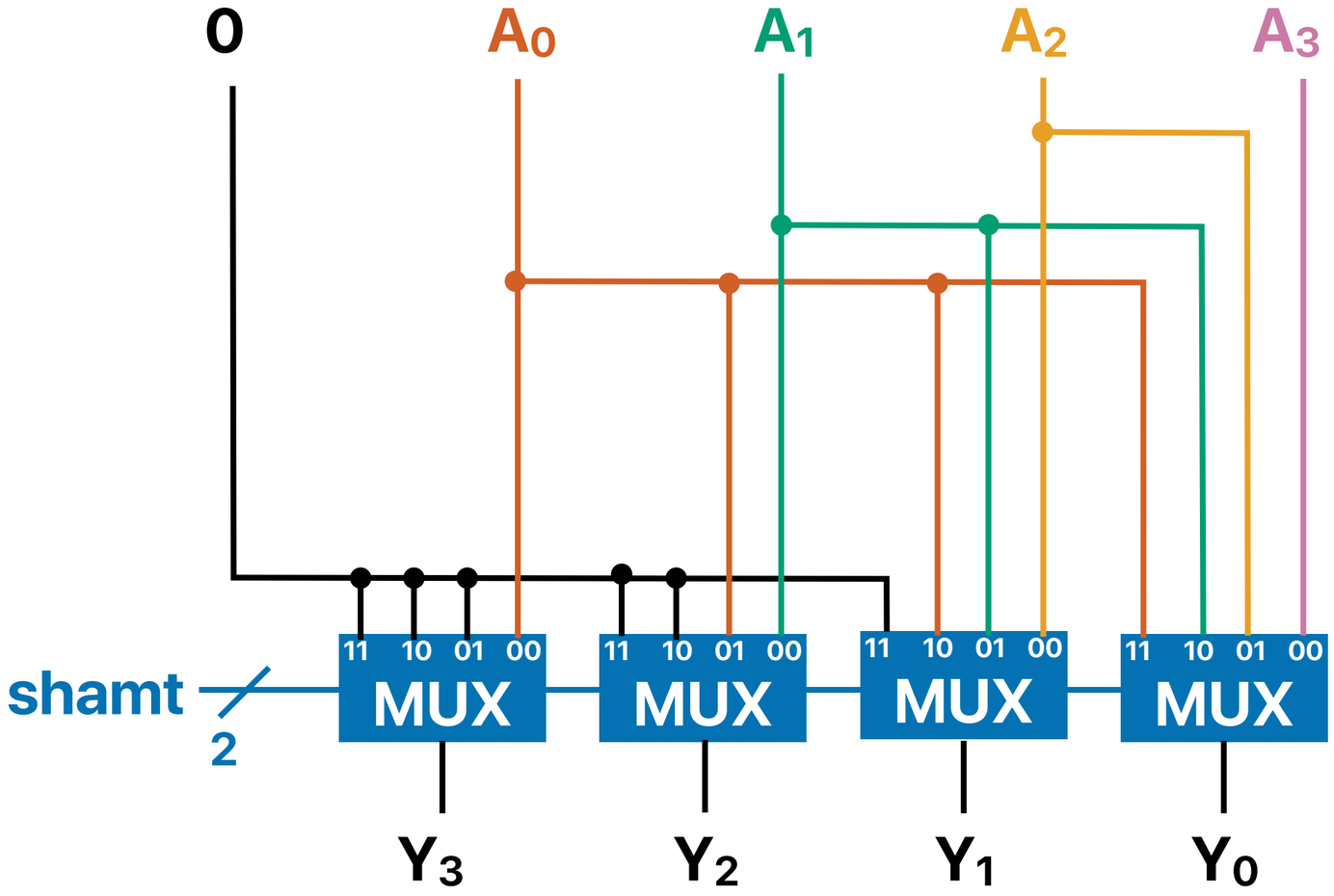


Shift “Left”

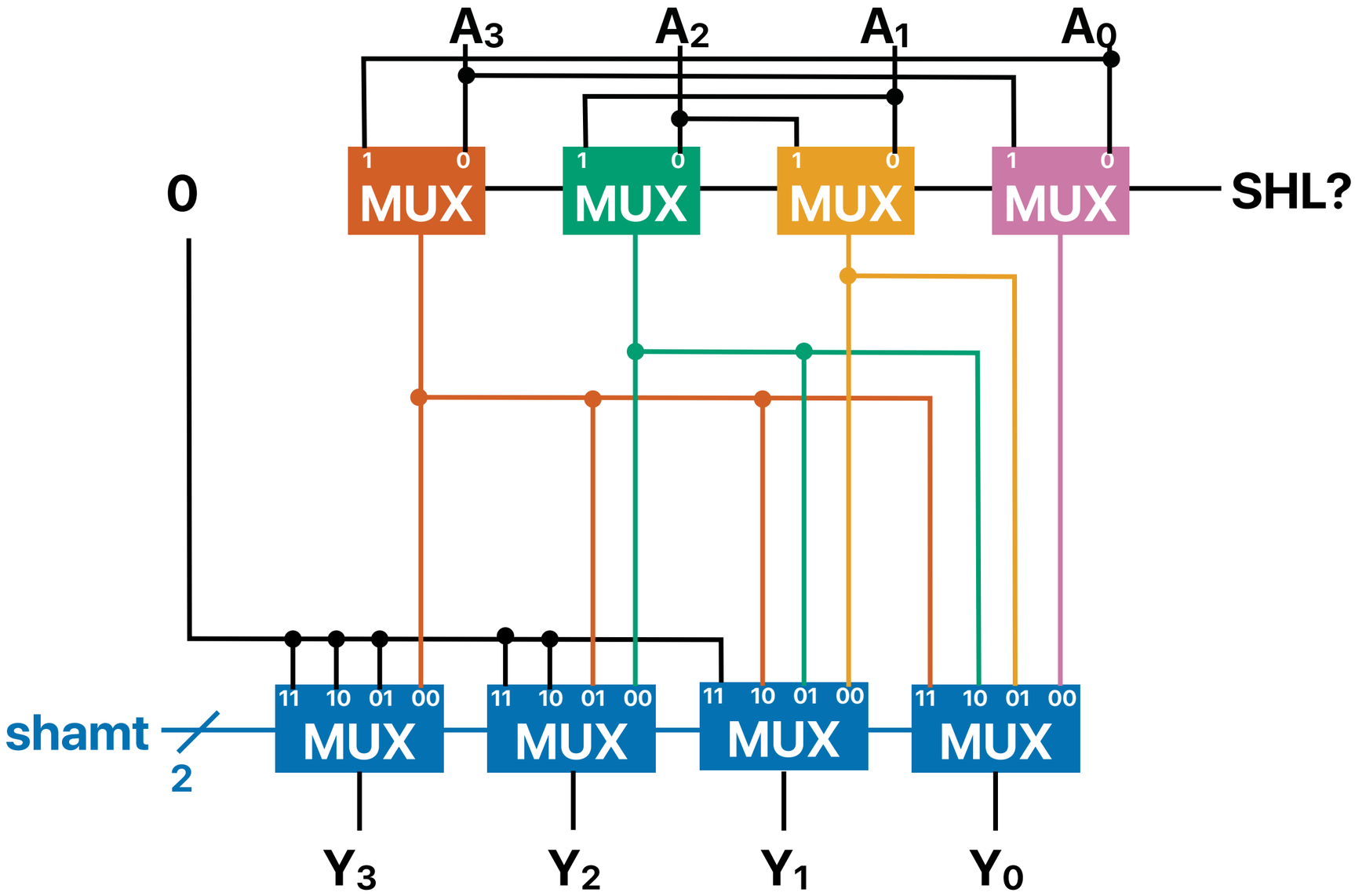
Example:
if S = 01
then
Y3 = A2
Y2 = A1
Y1 = A0
Y0 = 0

Example:
if S = 10
then
Y3 = A1
Y2 = A0
Y1 = 0
Y0 = 0

Example:
if S = 11
then
Y3 = A0
Y2 = 0
Y1 = 0
Y0 = 0

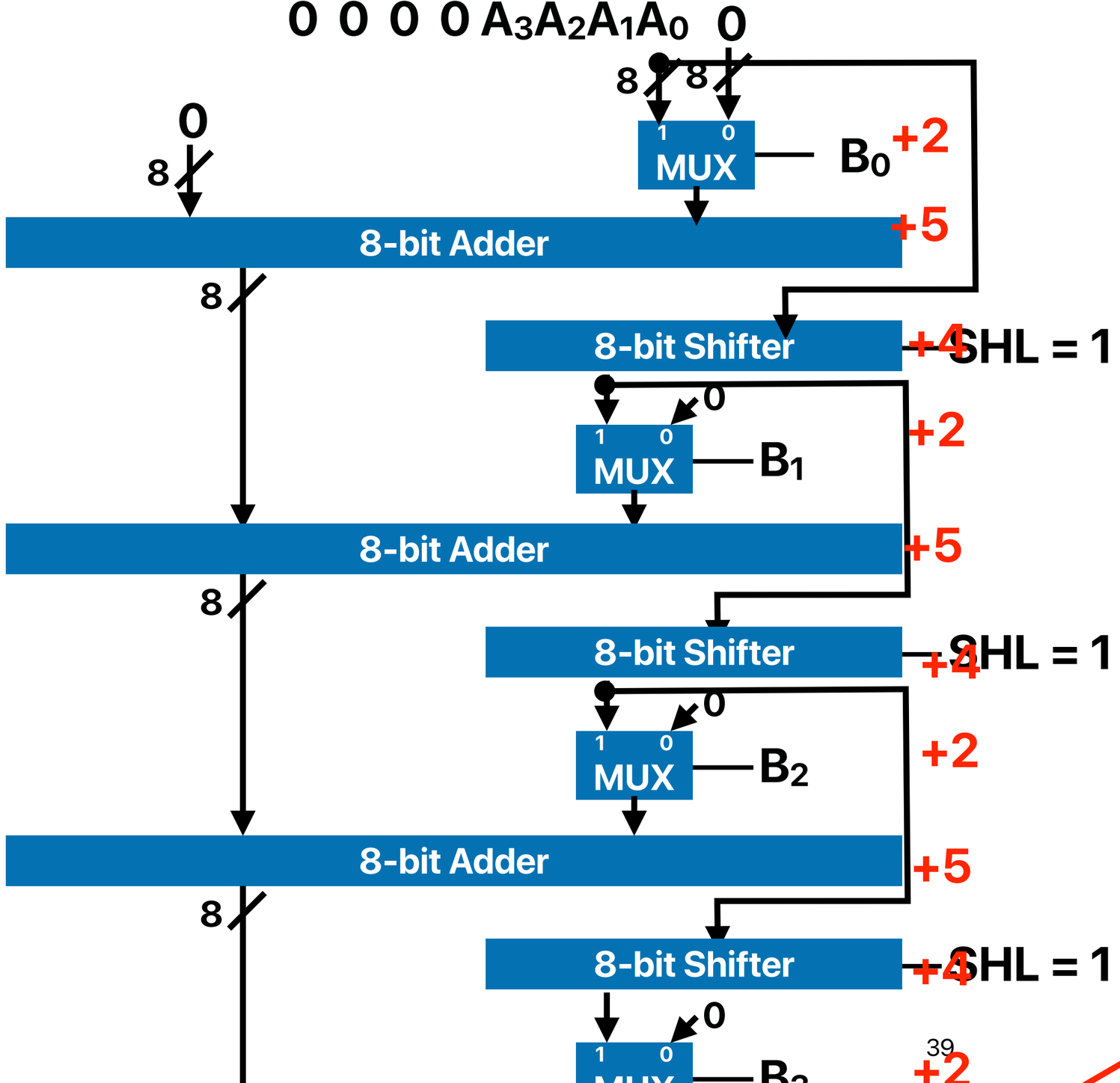


Generic Shifter



Let’s get back on Multiplier

Shift and add

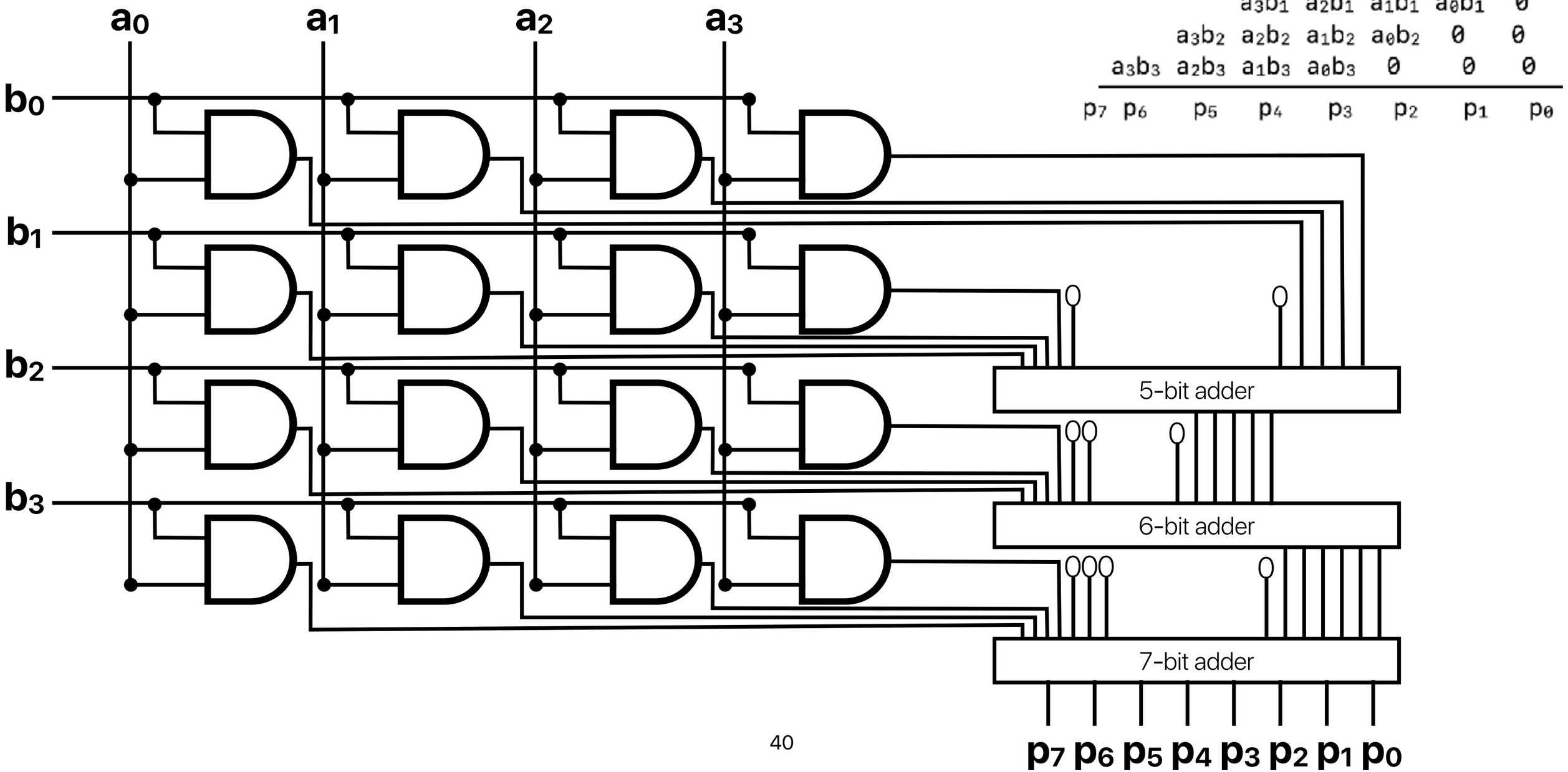


				a_3	a_2	a_1	a_0
				× b_3	b_2	b_1	b_0
				a_3b_0	a_2b_0	a_1b_0	a_0b_0
			a_3b_1	a_2b_1	a_1b_1	a_0b_1	0
		a_3b_2	a_2b_2	a_1b_2	a_0b_2	0	0
	a_3b_3	a_2b_3	a_1b_3	a_0b_3	0	0	0
p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

— 40 gate delays

Array style

			a_3	a_2	a_1	a_0	
			× b_3	b_2	b_1	b_0	
			a_3b_0	a_2b_0	a_1b_0	a_0b_0	
		a_3b_1	a_2b_1	a_1b_1	a_0b_1	0	
	a_3b_2	a_2b_2	a_1b_2	a_0b_2	0	0	
a_3b_3	a_2b_3	a_1b_3	a_0b_3	0	0	0	
p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

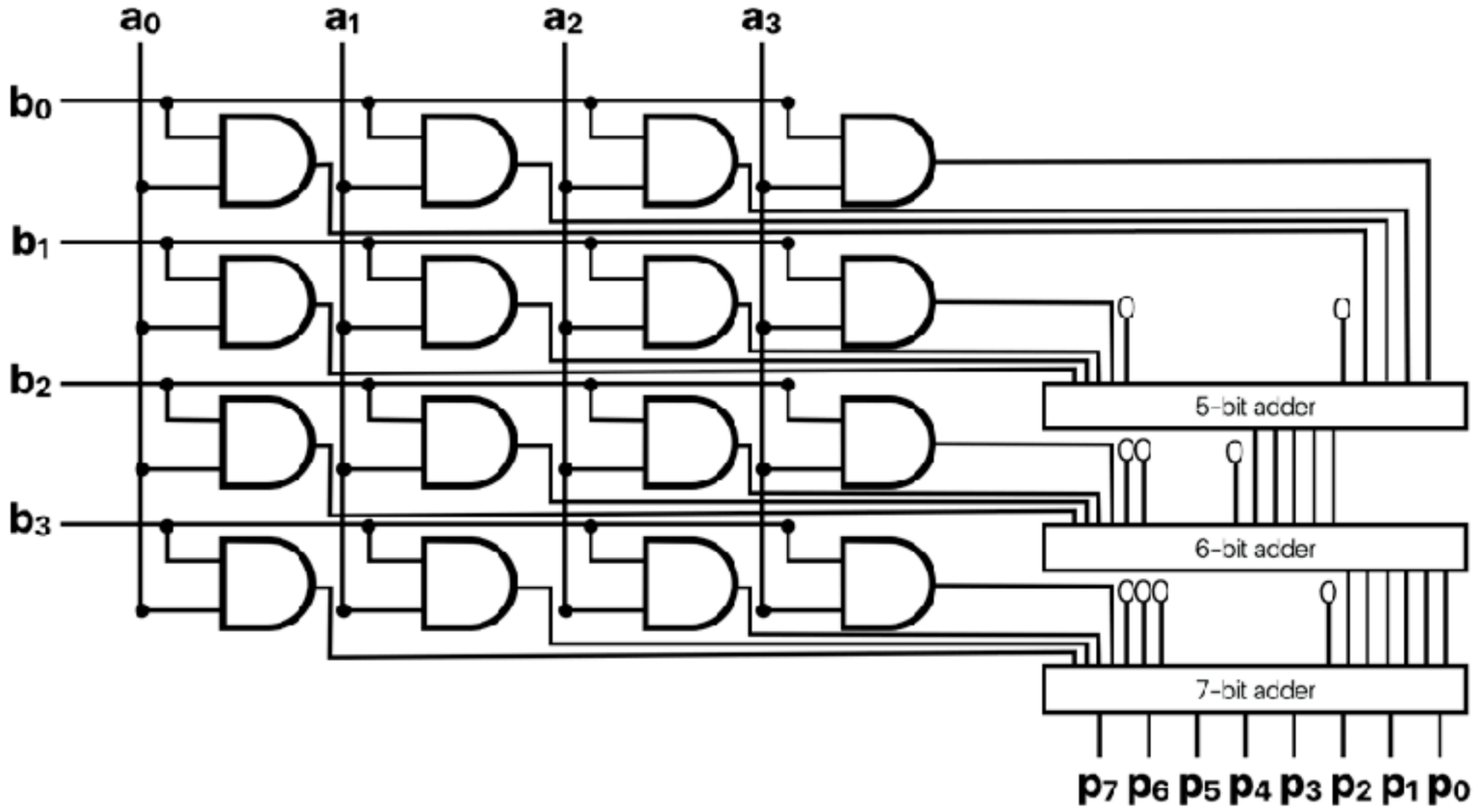


Poll close in 1:30

Gate-delays of Array-style Multipliers

- What's the estimated gate-delay of the 4-bit multiplier? (Assume adders are composed of 4-bit CLAs)

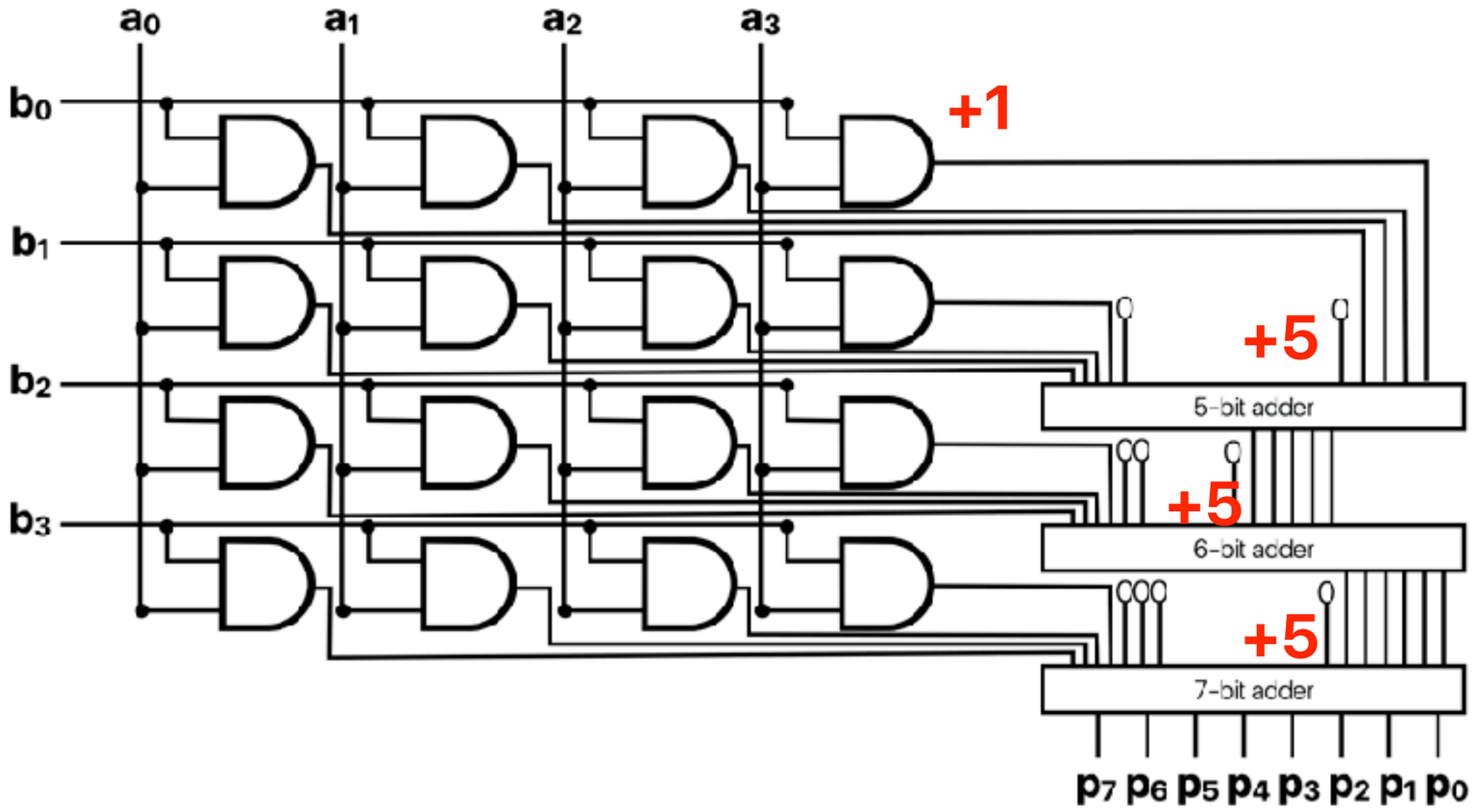
A. 9
B. 12
C. 13
D. 15
E. 16



Gate-delays of Array-style Multipliers

- What's the estimated gate-delay of the 4-bit multiplier? (Assume adders are composed of 4-bit CLAs)

A. 9
B. 12
C. 13
D. 15
E. 16



Poll close in 1:30

Gate-delays of 32-bit array-style multipliers

- What's the estimated gate-delay of a 32-bit multiplier? (Assume adders are composed of 4-bit CLAs)
  A. 0 — 100
  B. 100 — 500
  C. 500 — 1000
  D. 1000 — 1500
  E. > 1500

Gate-delays of 32-bit array-style multipliers

- What's the estimated gate-delay of a 32-bit multiplier? (Assume adders are composed of 4-bit CLAs)
  - A. 0 — 100
  - B. 100 — 500
  - C. 500 — 1000
  - D. 1000 — 1500
  - E. > 1500

Each n-bit adder is roundup(n/4)*2+1

We need 33-64 bit adders

33 - 36 -bit adders —> (9*2+1) gate delays *4

37 - 40 -bit adders —> (10*2+1) gate delays *4

41 - 44 -bit adders —> (11*2+1) gate delays *4

45 - 48 -bit adders —> (12*2+1) gate delays *4

49 - 52 -bit adders —> (13*2+1) gate delays *4

53 - 56 -bit adders —> (14*2+1) gate delays *4

57 - 60 -bit adders —> (15*2+1) gate delays *4

61 - 64 -bit adders —> (16*2+1) gate delays *4

4*2*(9+10+11+12+13+14+15+16+1) = 808

Announcement

- Lab 5 due tonight
- Lab 6 is up — due on 6/2
  - Watch the video and read the instruction BEFORE your session
  - There are links on both course webpage and iLearn lab section
  - Submit through iLearn > Labs
- Office Hours
  - All office hours share the same meeting instance — if you have registered once, you cannot do it again.
  - Zoom does not resend registration confirmation and does not allow us to “re-approve” if you have registered
  - The only way is to dig out the e-mail from Zoom
- Last reading quiz due next Tuesday
- Check your grades in iLearn

Electrical
Computer Science
Engineering

120A

つづく