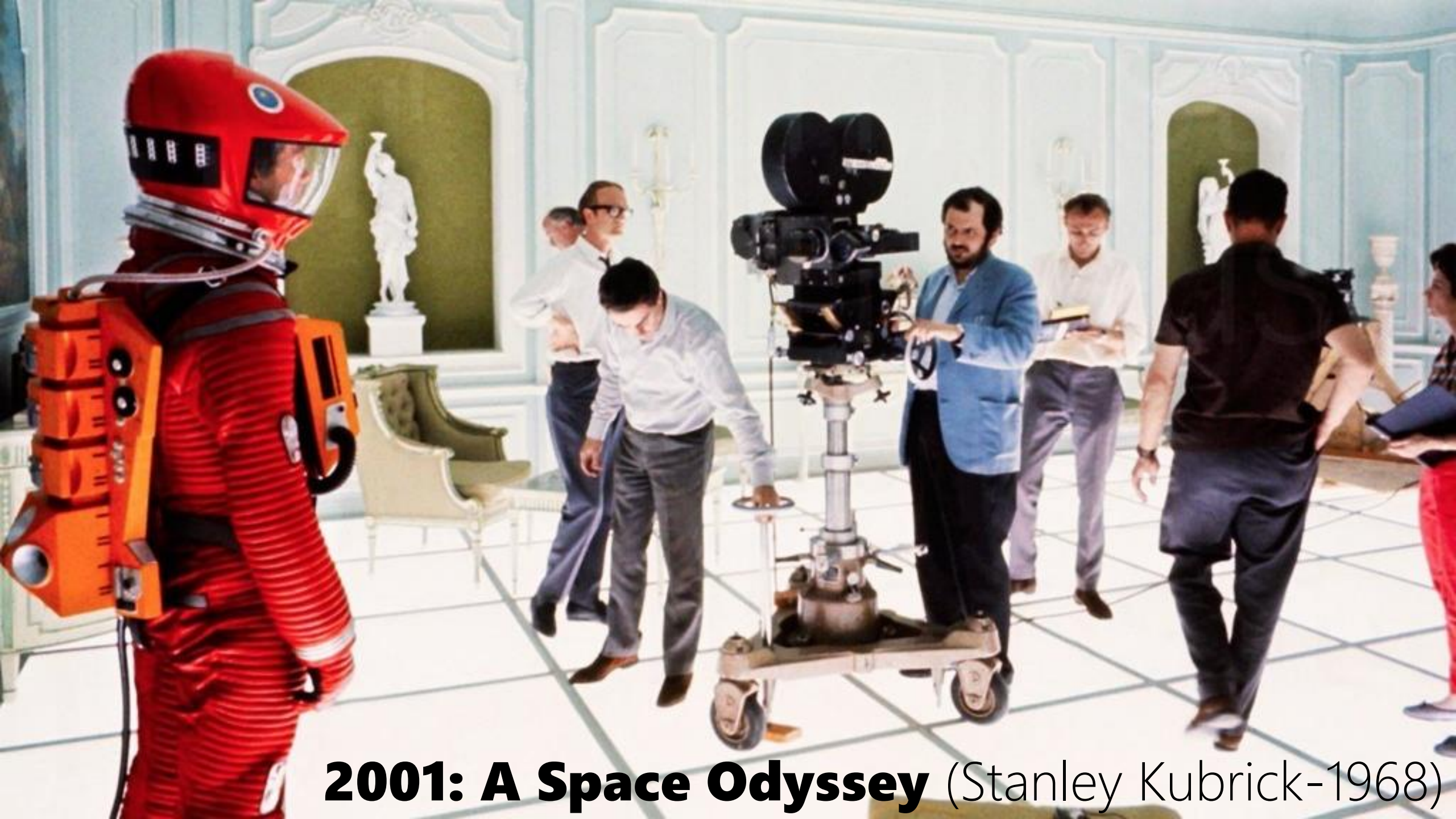
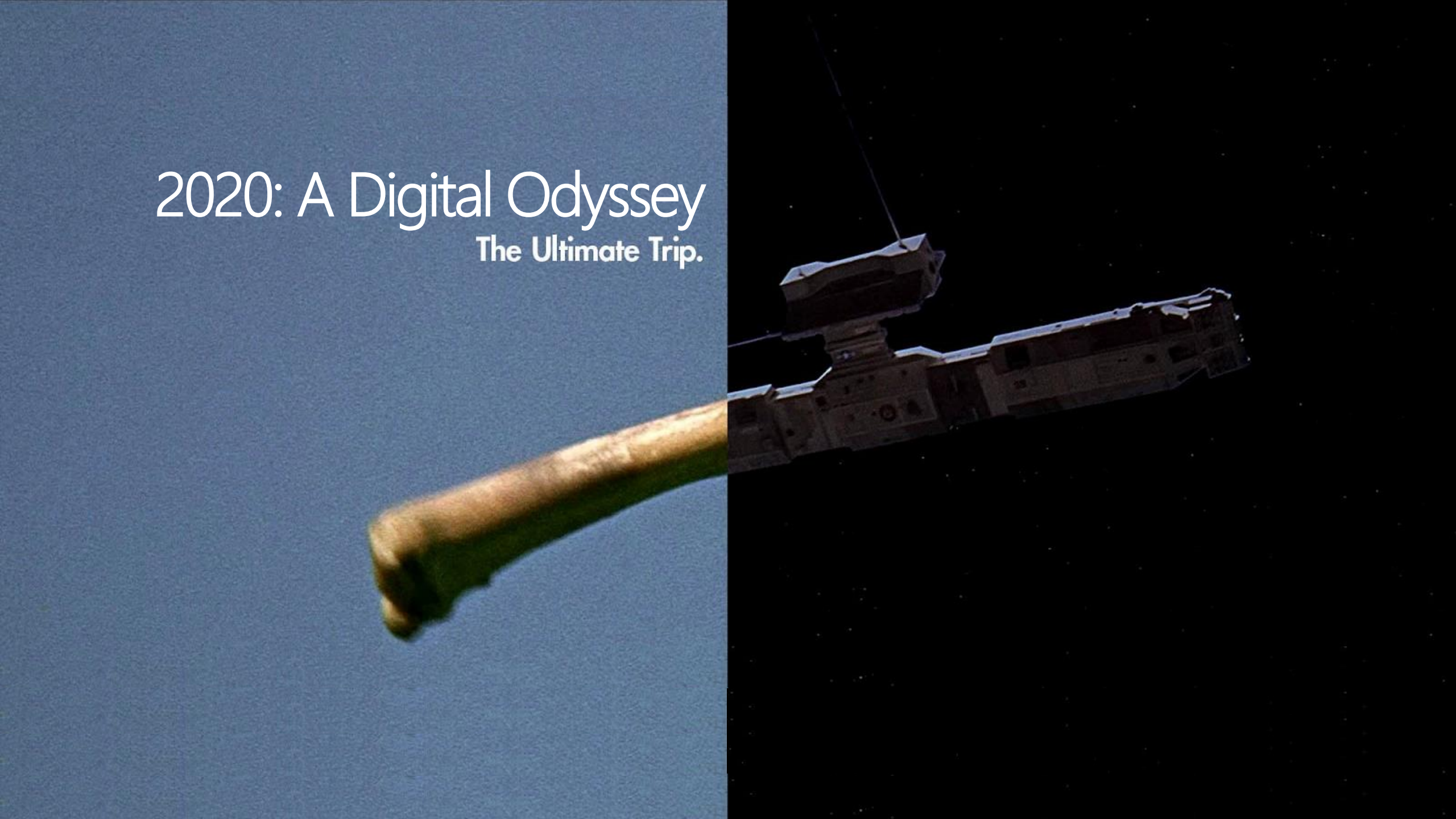


2001: A Space Odyssey (Stanley Kubrick-1968)

2020: A Digital Odyssey

The Ultimate Trip.



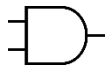
Number Systems |

| Computer Systems

Number Systems | $(12)_{10} \rightarrow (1100)_2$

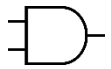
| Computer Systems

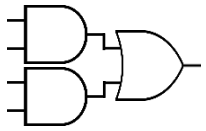
Number Systems | $(12)_{10} \rightarrow (1100)_2$

| Logic Gates | 

| Computer Systems

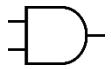
Number Systems | $(12)_{10} \rightarrow (1100)_2$

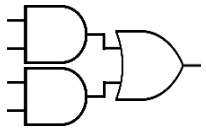
| Logic Gates | 

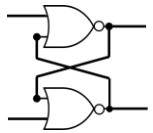
| Combinational Logic | 

| Computer Systems

Number Systems | $(12)_{10} \rightarrow (1100)_2$

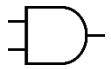
| Logic Gates | 

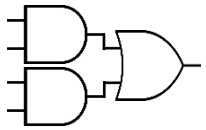
| Combinational Logic | 

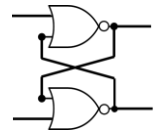
| Flip-Flops | 

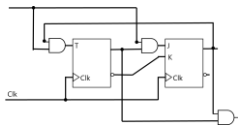
| Computer Systems

Number Systems | $(12)_{10} \rightarrow (1100)_2$

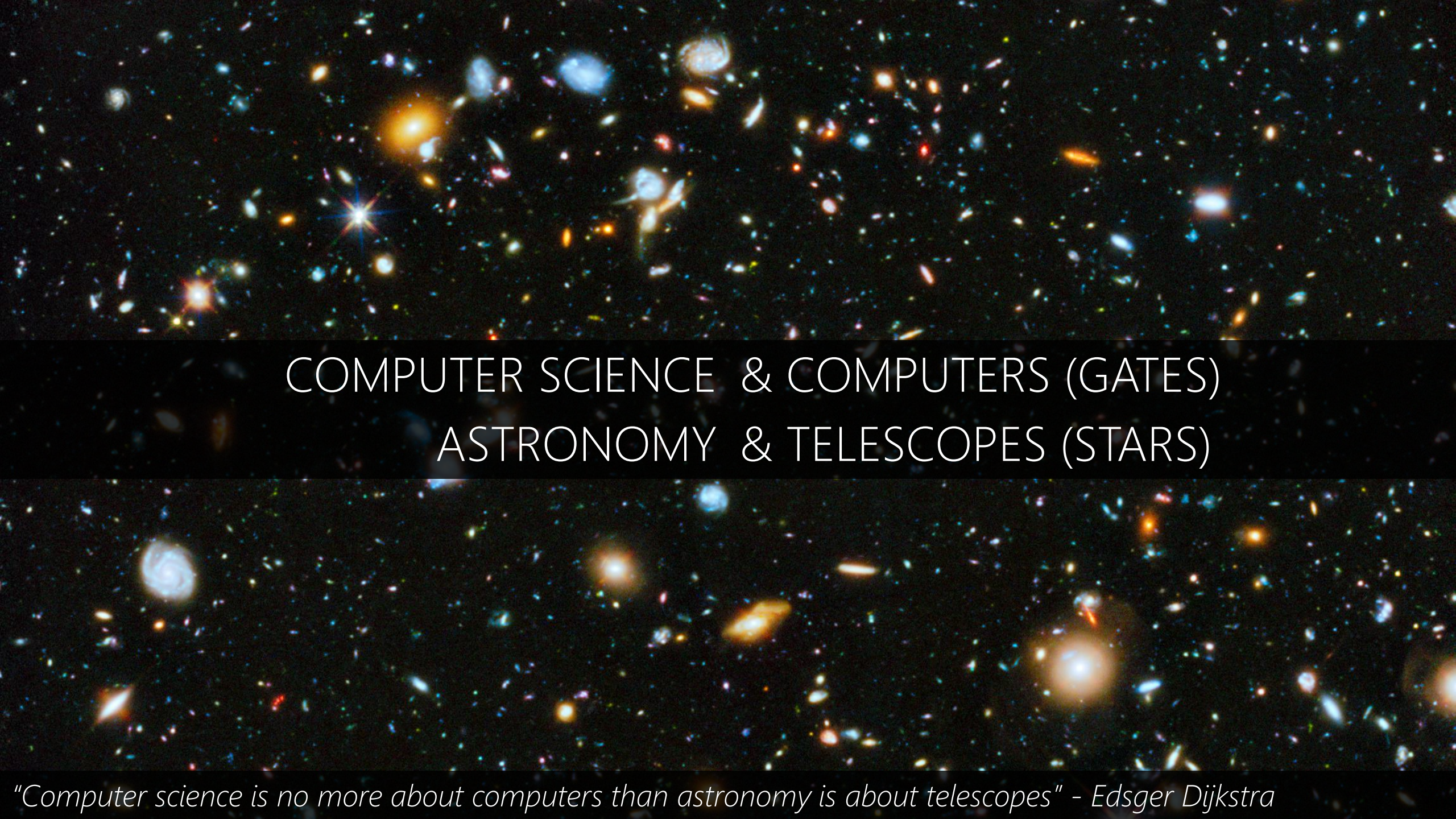
| Logic Gates | 

| Combinational Logic | 

| Flip-Flops | 

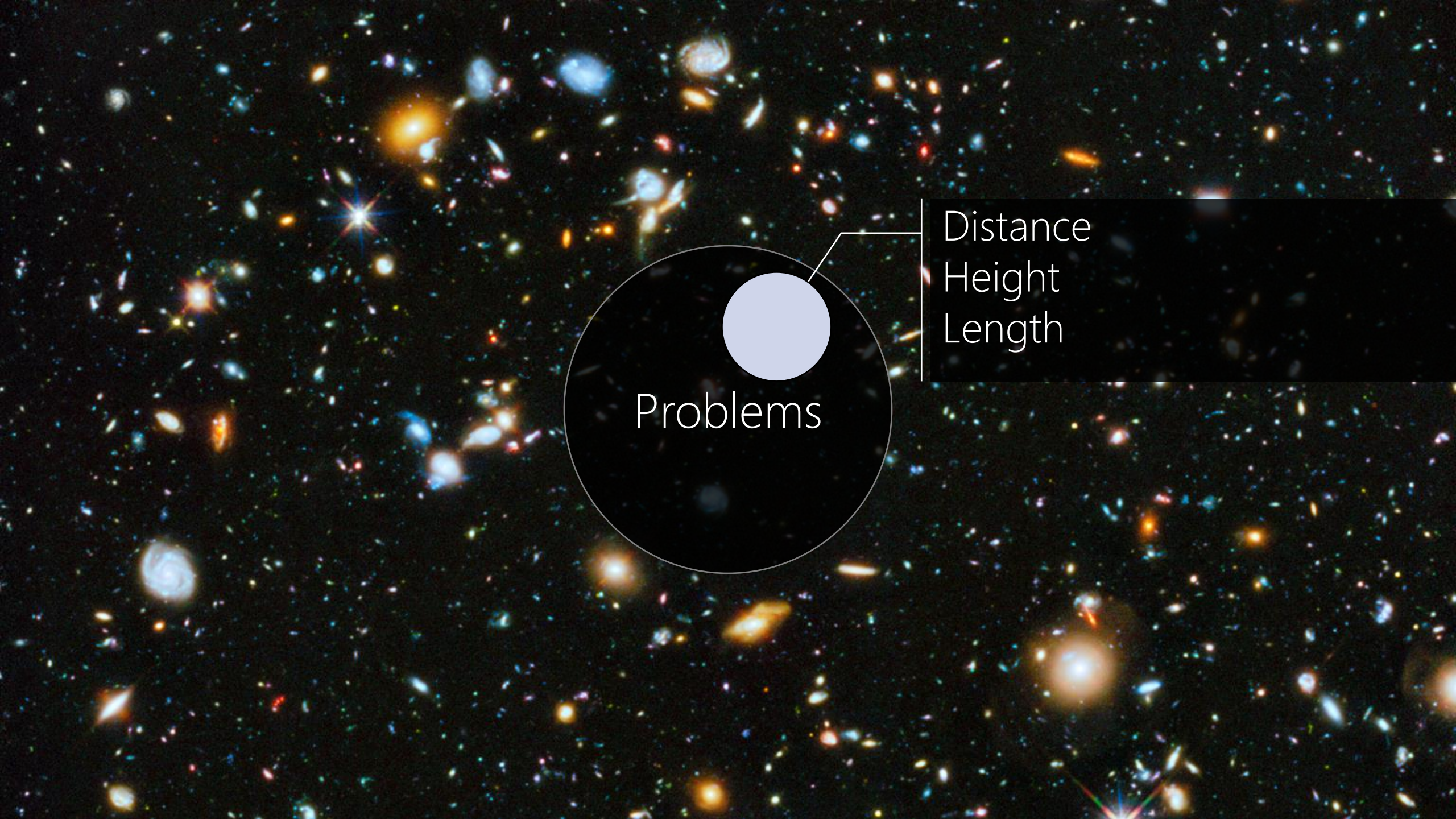
| Sequential Logic | 

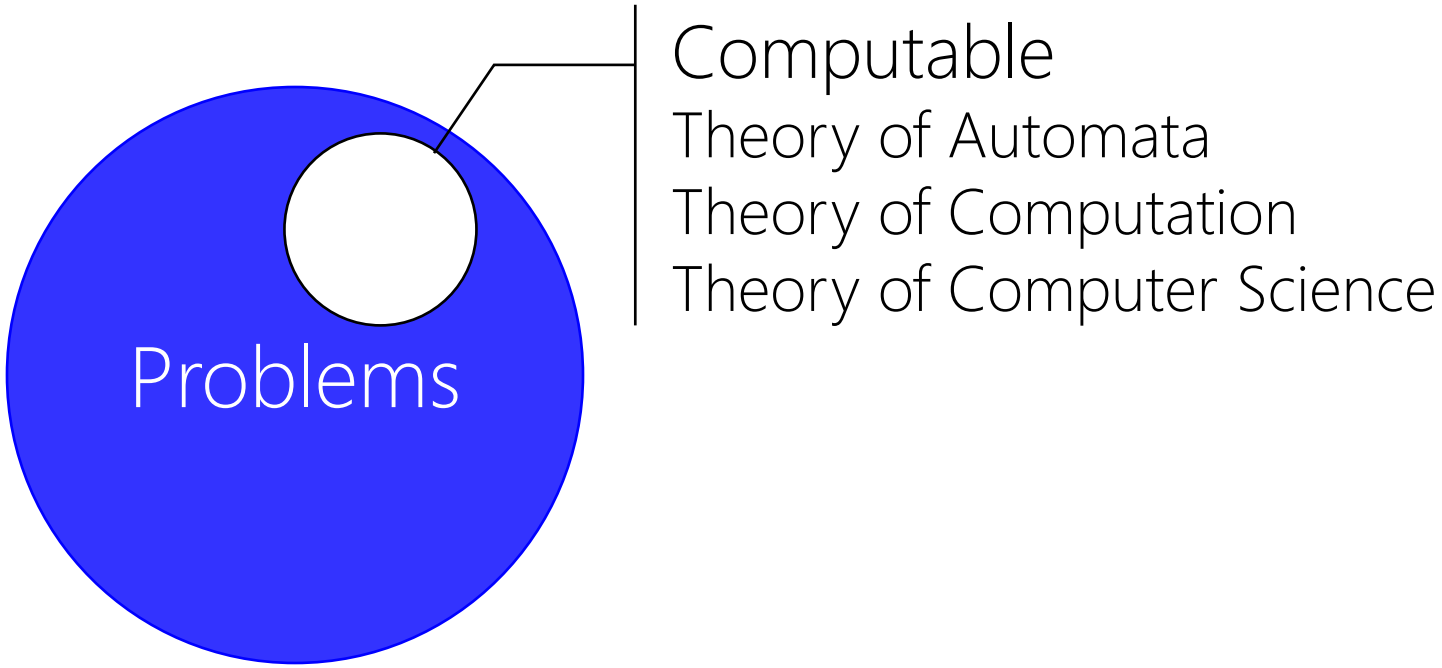
| Computer Systems

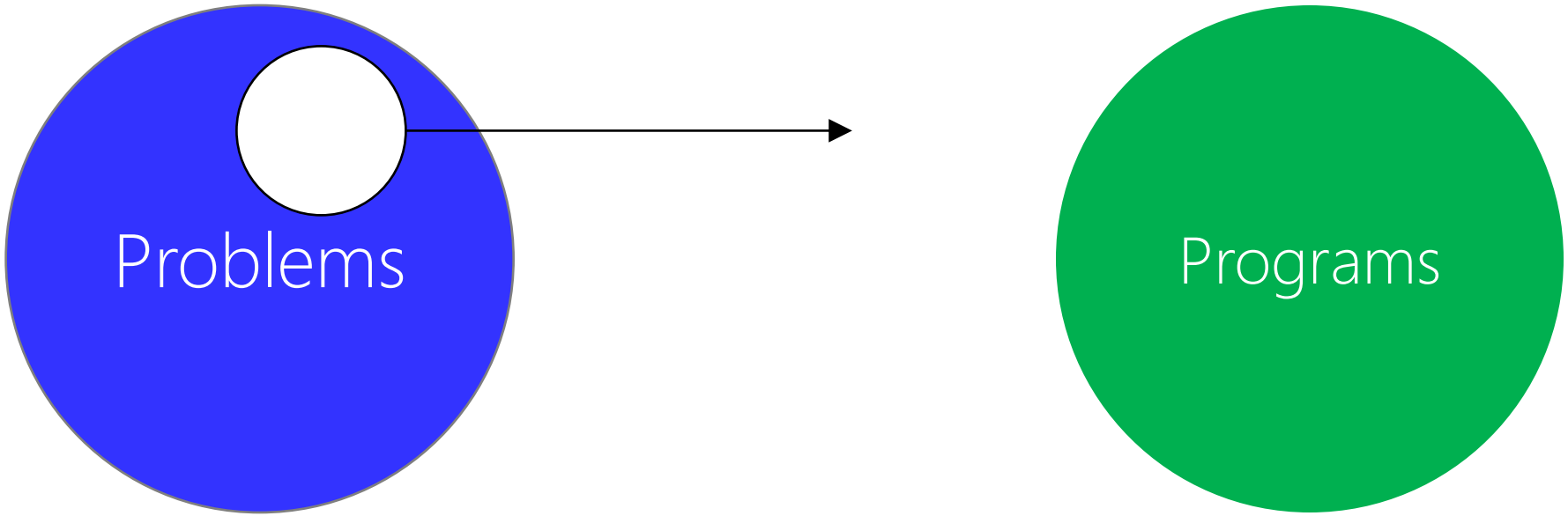


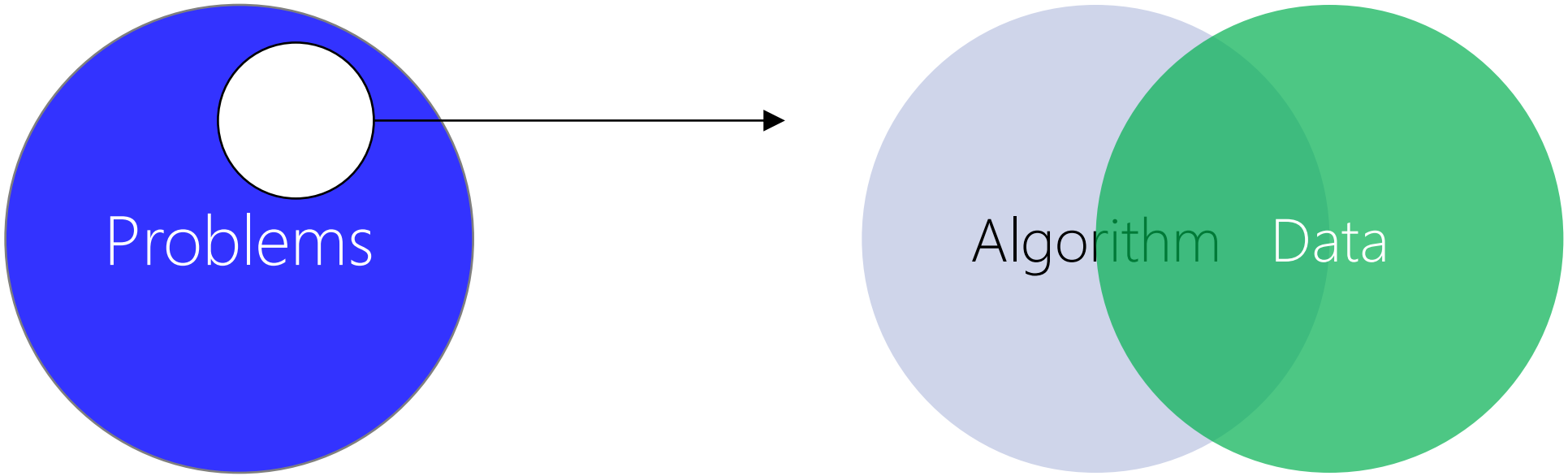
COMPUTER SCIENCE & COMPUTERS (GATES)
ASTRONOMY & TELESCOPES (STARS)

"Computer science is no more about computers than astronomy is about telescopes" - Edsger Dijkstra


Problems
Distance
Height
Length


Problems
Computable
Theory of Automata
Theory of Computation
Theory of Computer Science


Problems
Programs


Problems
Algorithm
Data

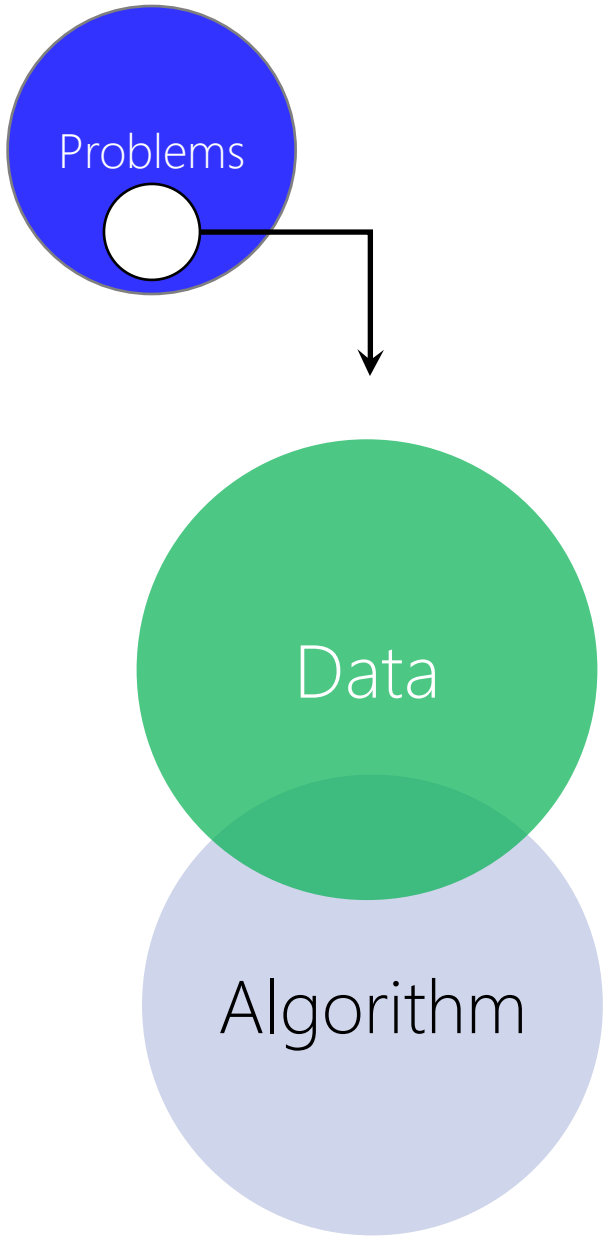
Design a Computer System

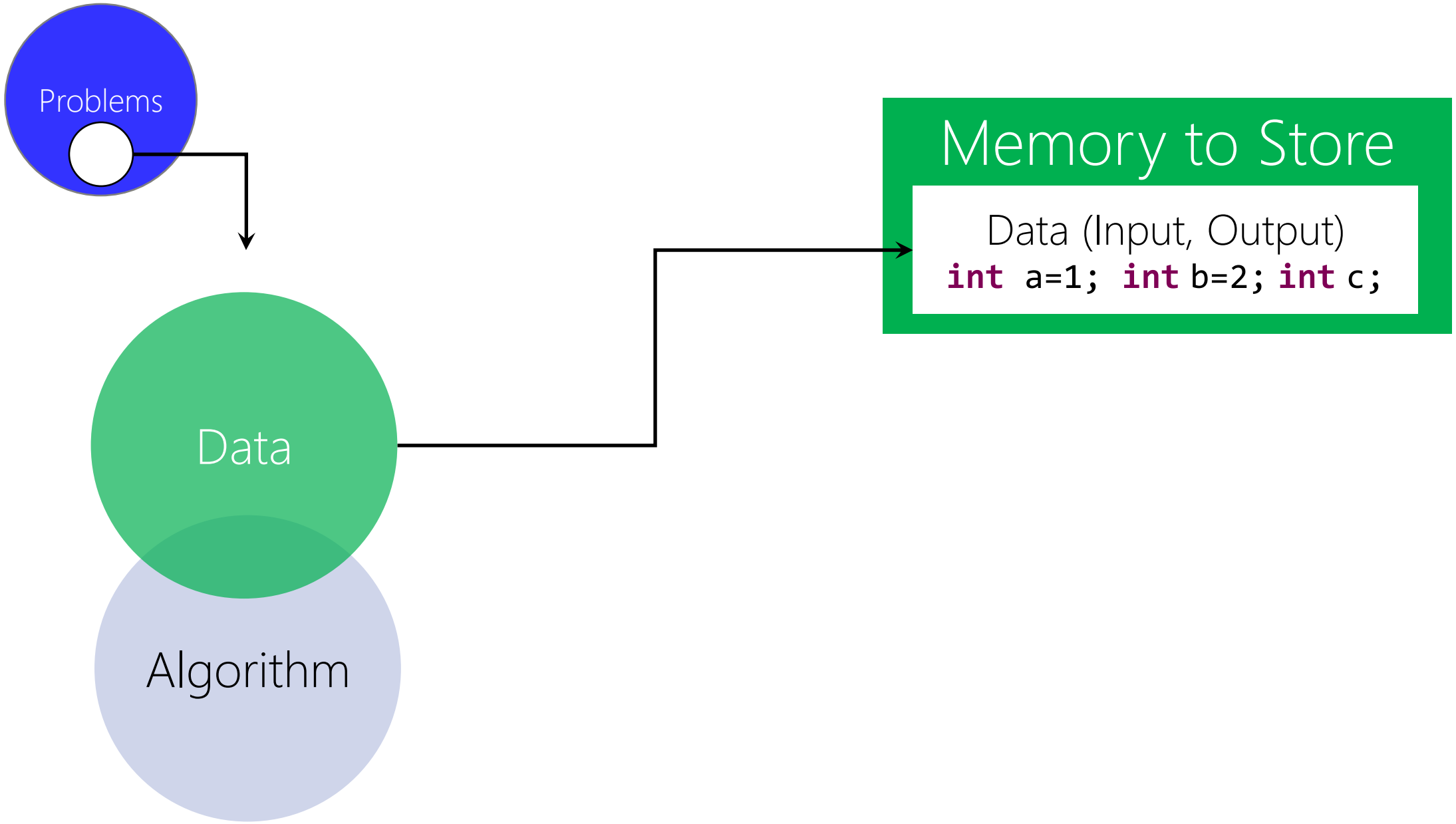
John von Neumann

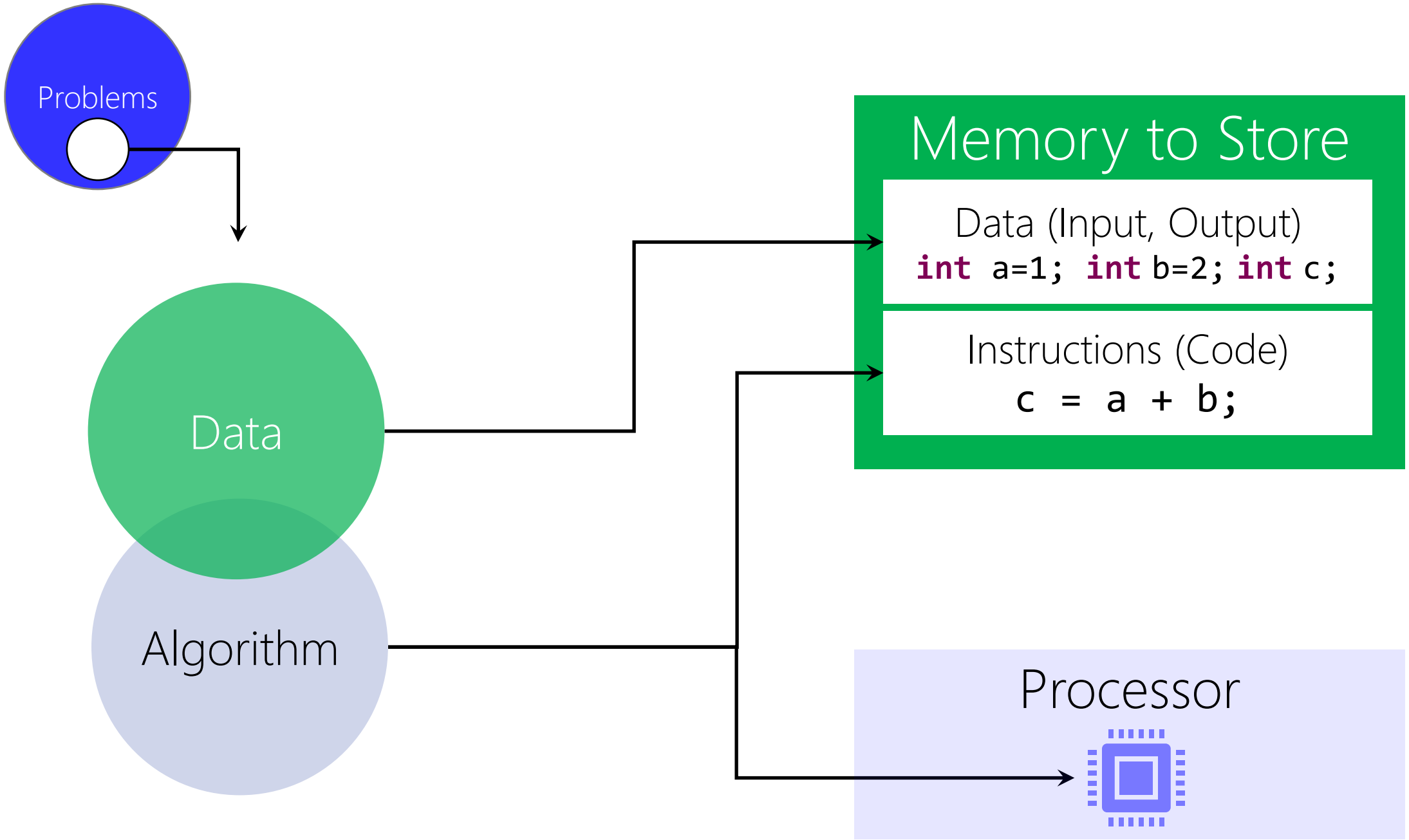
(/vɒn ˈnɔɪmən/)
1903 –1957
Mathematician, Physicist, Computer Scientist, Engineer

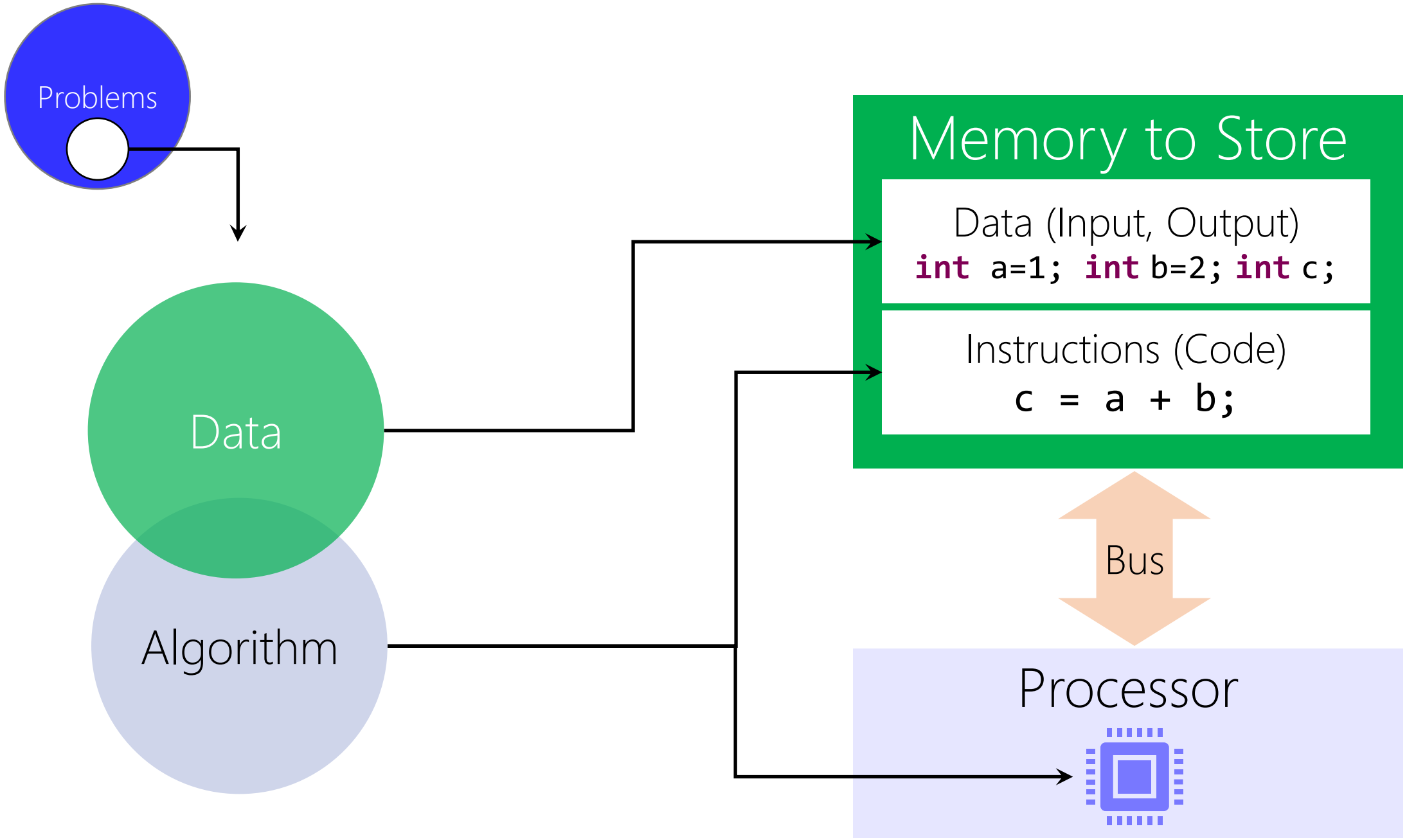
Polymath
He integrated pure and applied sciences. He made major contributions to many fields, including:

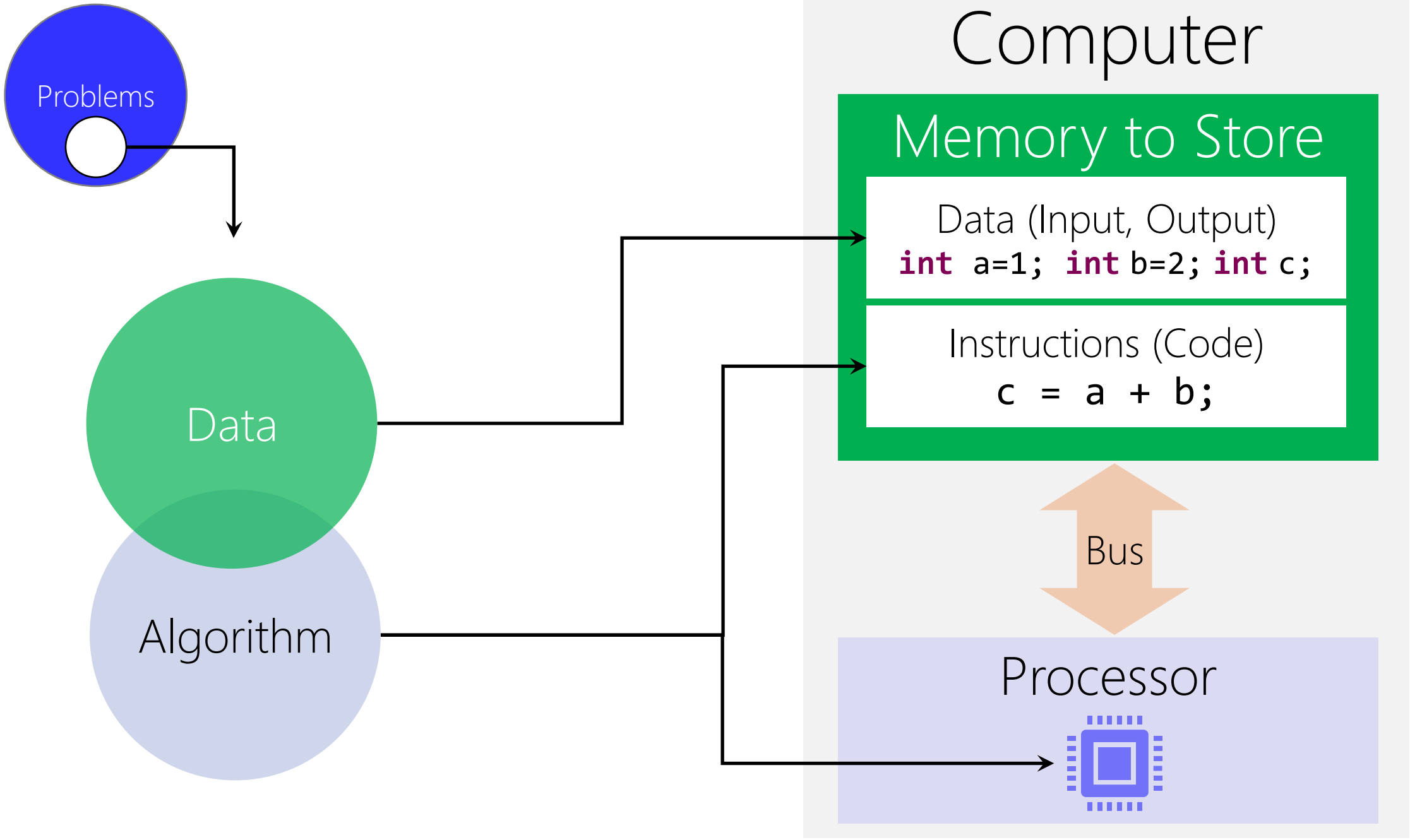
- Mathematics
- Physics
- Economics (game theory)
- Computing
- Statistics


Problems
Data
Algorithm


Problems
Data
Algorithm
Memory to Store
Data (Input, Output)
int a=1; int b=2; int c;


Problems
Data
Algorithm
Memory to Store
Data (Input, Output)
int a=1; int b=2; int c;
Instructions (Code)
c = a + b;
Processor


Problems
Data
Algorithm
Memory to Store
Data (Input, Output)
int a=1; int b=2; int c;
Instructions (Code)
c = a + b;
Bus
Processor


Problems
Data
Algorithm
Computer
Memory to Store
Data (Input, Output)
int a=1; int b=2; int c;
Instructions (Code)
c = a + b;
Bus
Processor

von Neumann Architecture

Principles

- Data and instructions are both stored in the main memory
- The content of the memory is addressable by location (regardless of what is stored in that location)
- Instructions are executed sequentially unless the order is explicitly modified

Computer System

Computer

Memory to Store

Data (Input, Output)

```
int a=1; int b=2; int c;
```

Instructions (Code)

```
c = a + b;
```

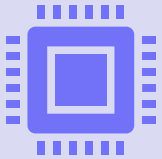
Input/Output Devices

```
scanf("%d", &a);
scanf("%d", &b);
printf("%d", c);
```

Bus

Bus

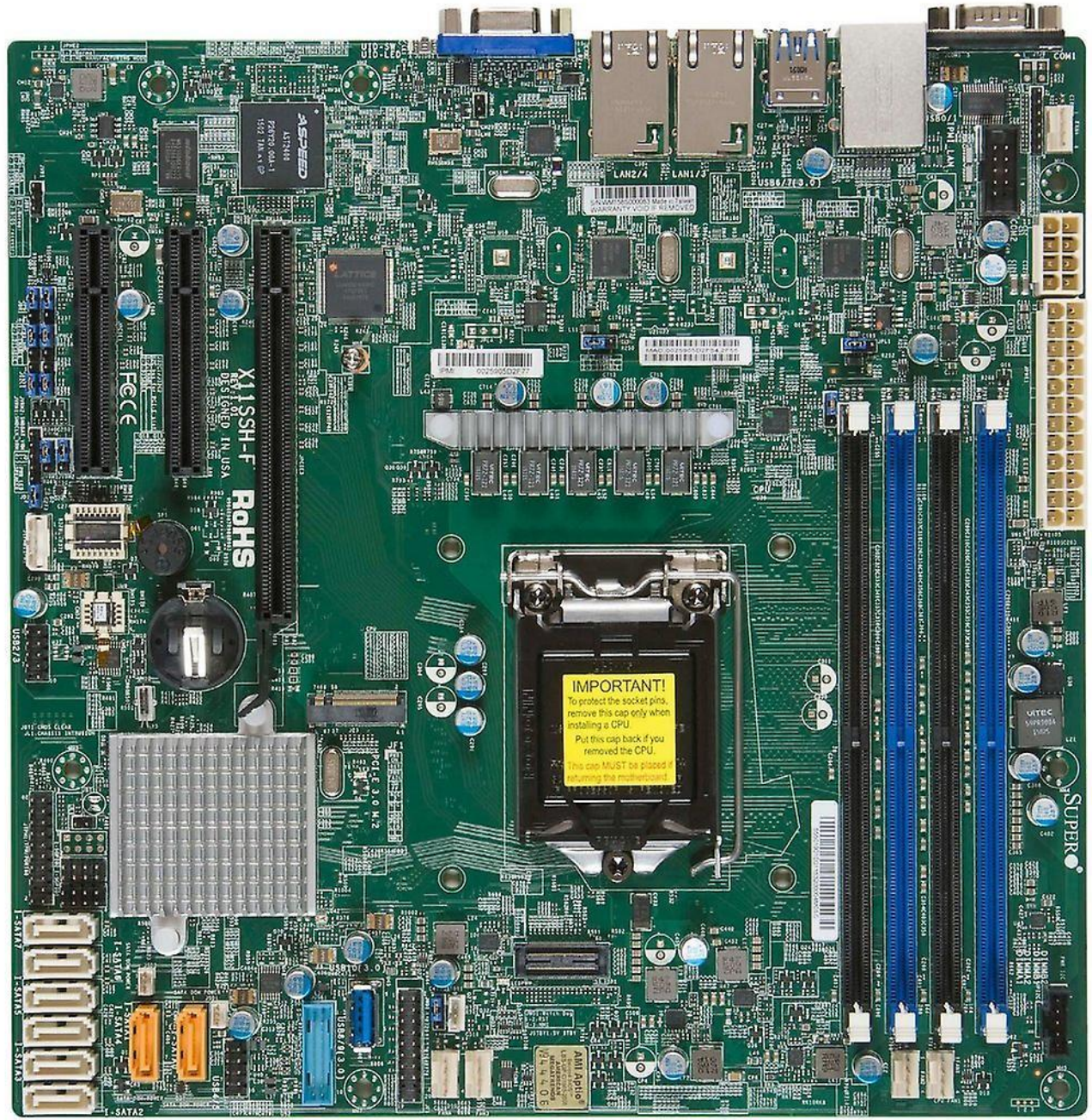
Processor

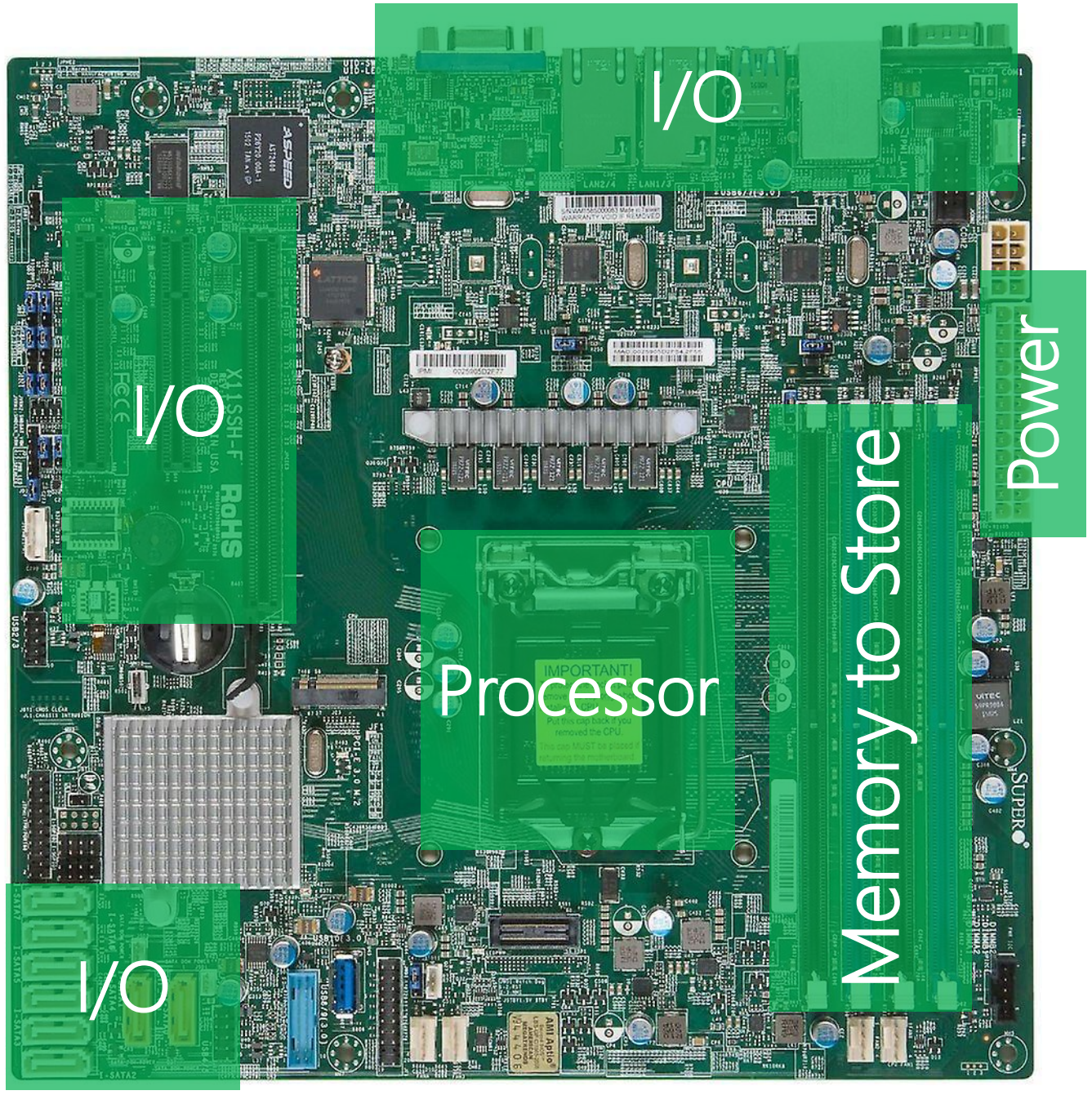


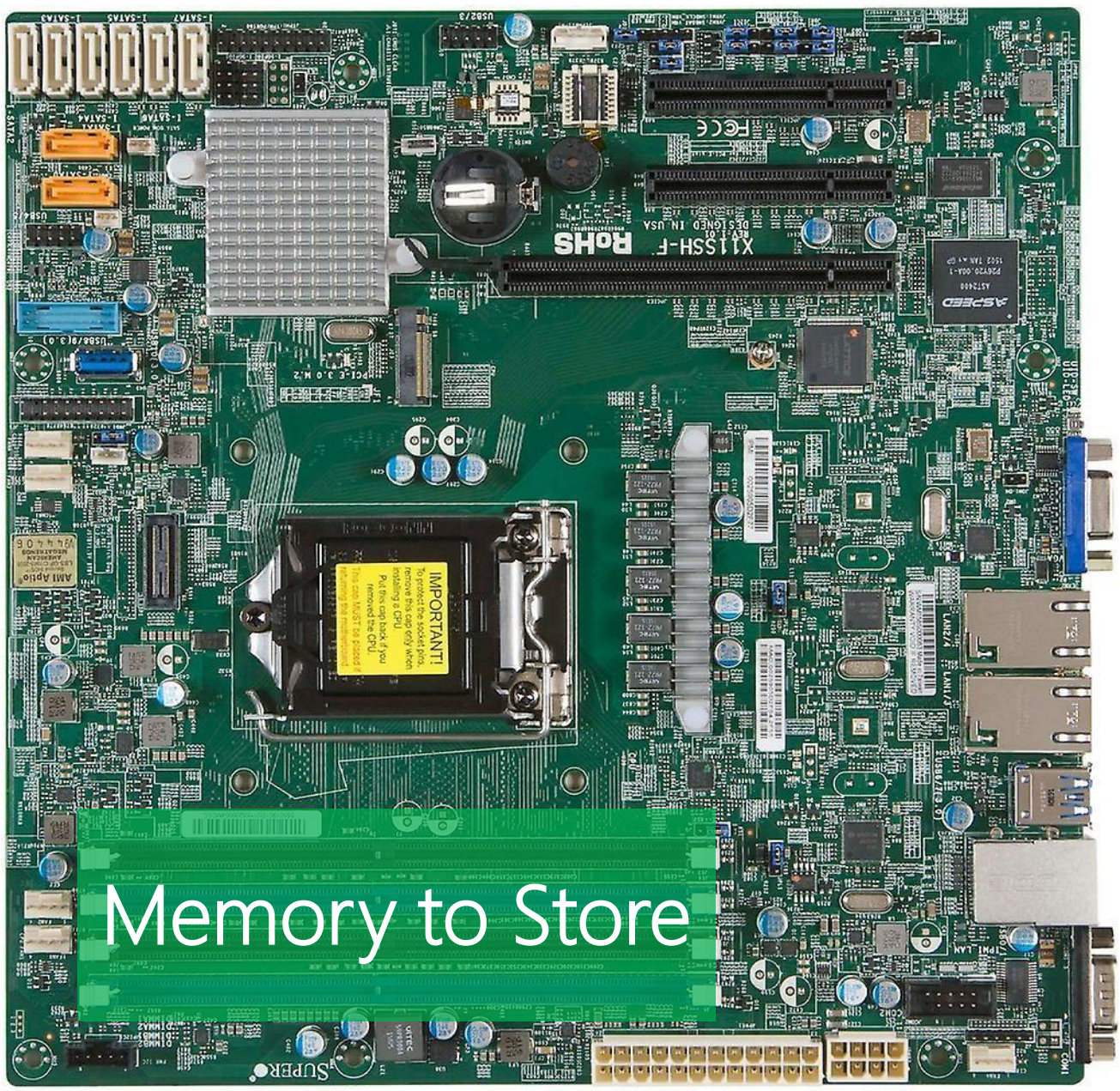
Bus

Permanent Storage

```
fprintf()
fscanf()
fread()
fwrite()
fseek()
```


X11SSH-F
Rev. 1.01
DESIGNED IN USA
RoHS
IMPORTANT!
To protect the socket pins, remove this cap only when installing a CPU.
Put this cap back if you removed the CPU.
This cap MUST be placed if returning the motherboard.
LAN2/4
LAN1/3
USB6/7(3.0)
COM1
SUPER


I/O
I/O
Power
Memory to Store
Processor
I/O



Memory to Store

Design Memory

To Store Data + Code

Design Memory

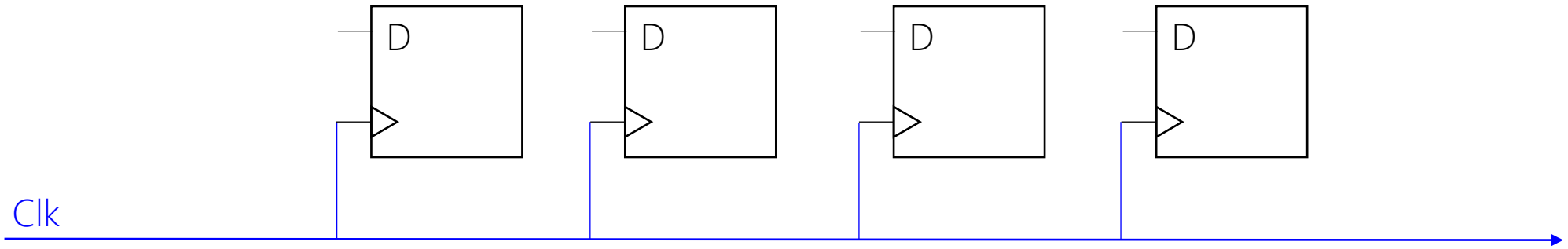
Data → Numbers

Design Memory

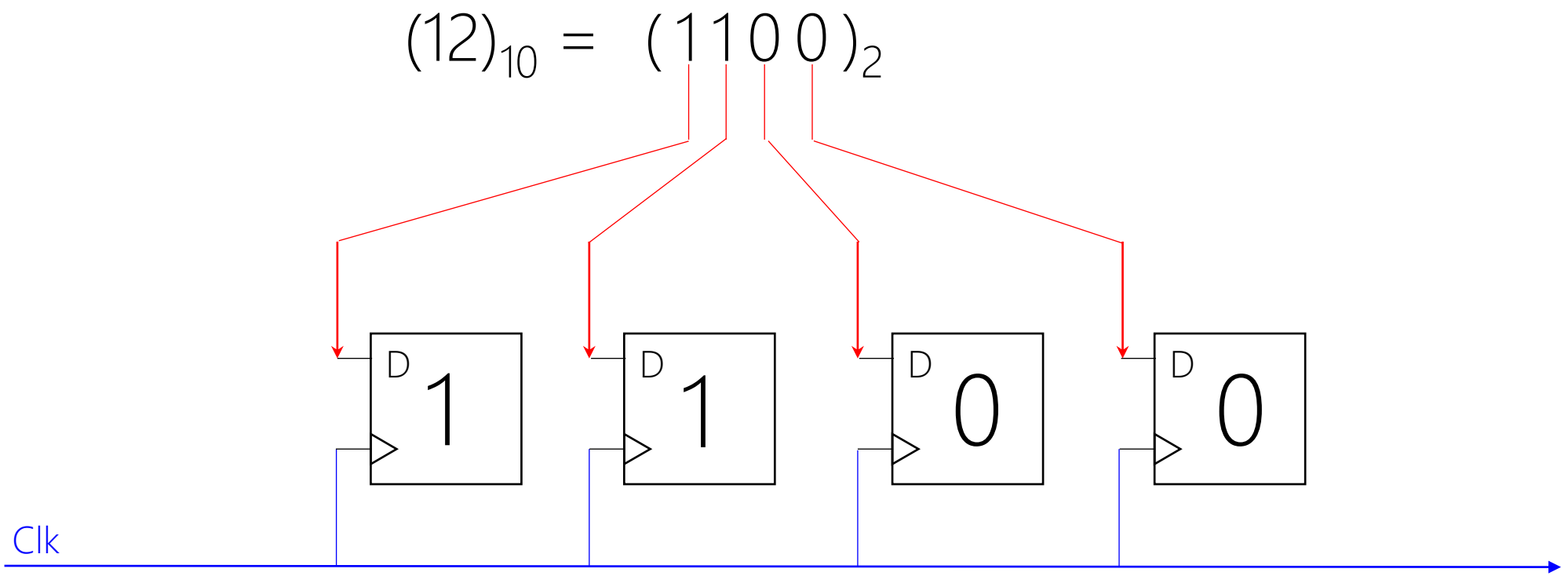
Each Number $\in$ [min, max] $\rightarrow$ Binary System

Design Memory

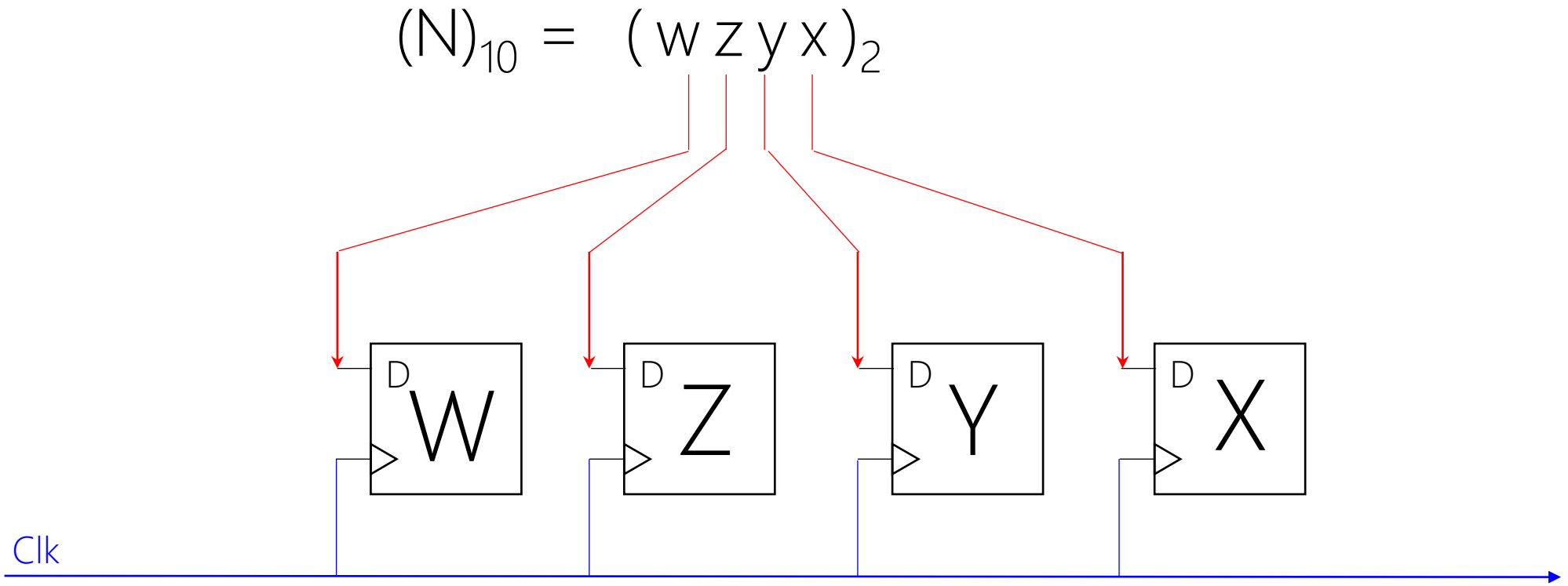
Each Number $\in [0, 2^n) \rightarrow$ n Flip-Flops



n=4 ⟶ Each Number ∈ [0, 2^4=16) ⟶ 4 Flip-Flops

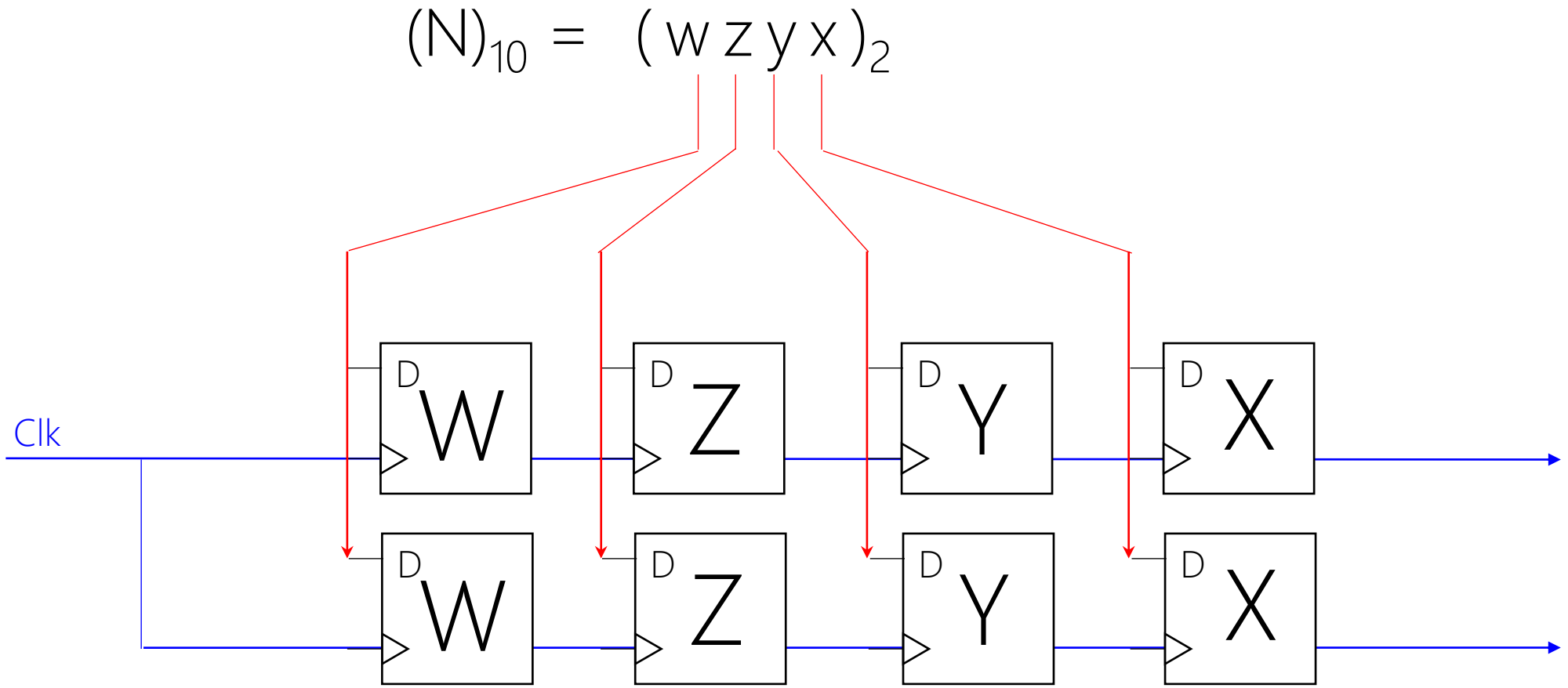


n=4 ⟶ Each Number ∈ [0, 2^4=16) ⟶ 4 Flip-Flops

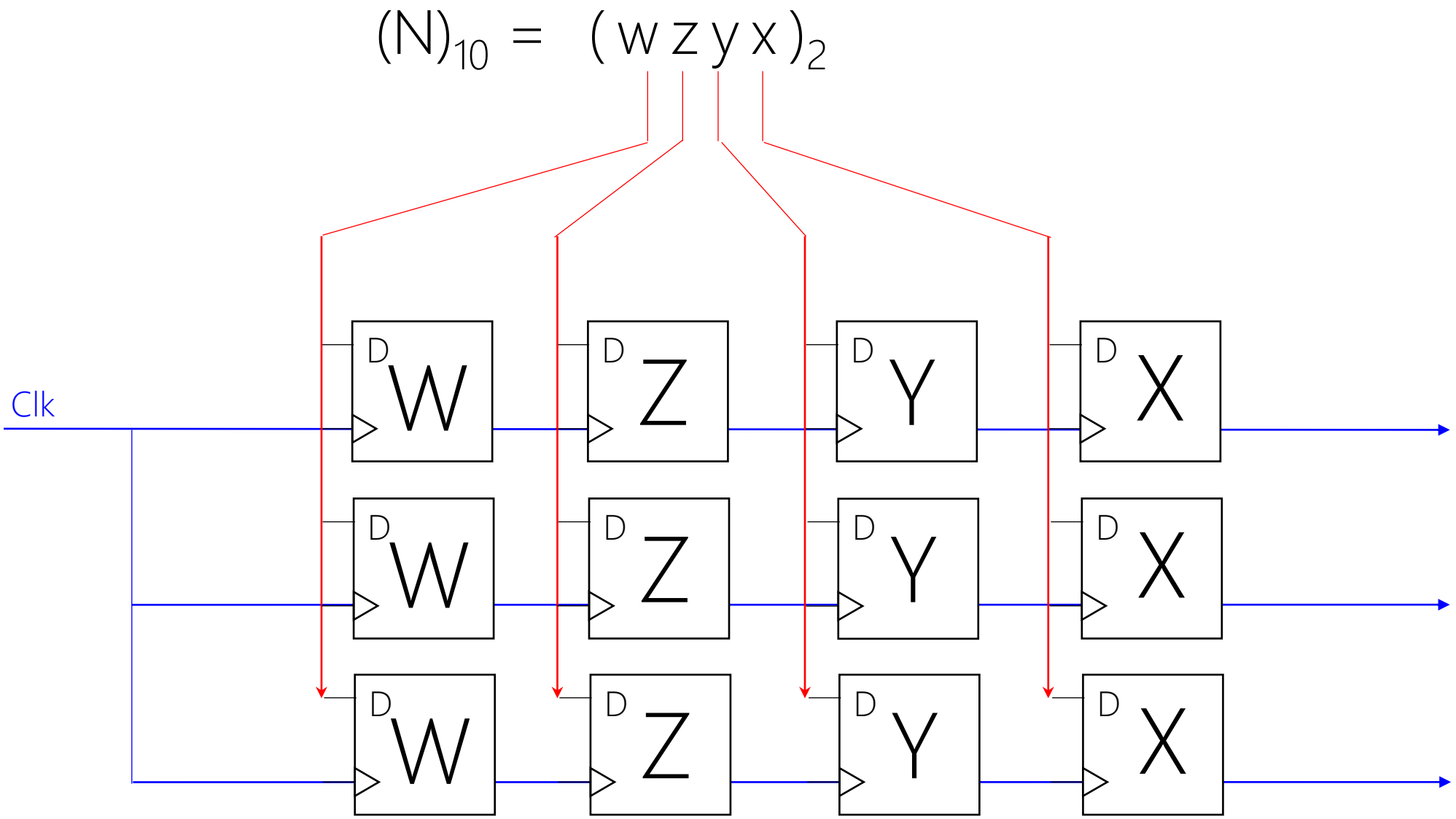


1 × Number ⟶ 1 × n FFs ⟶ 1 × n-bit Register

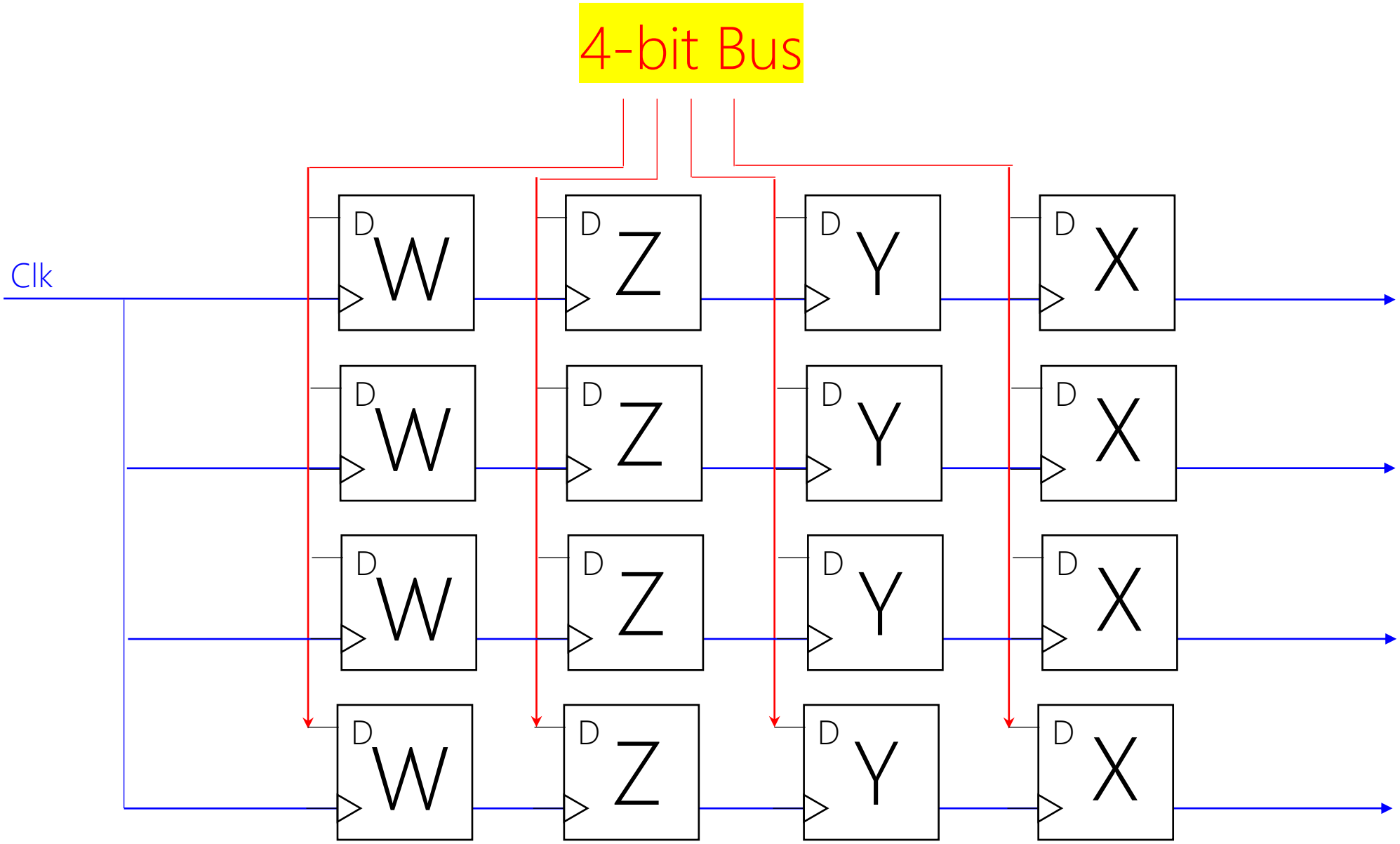
Register ⟶ an vector of flip-flops



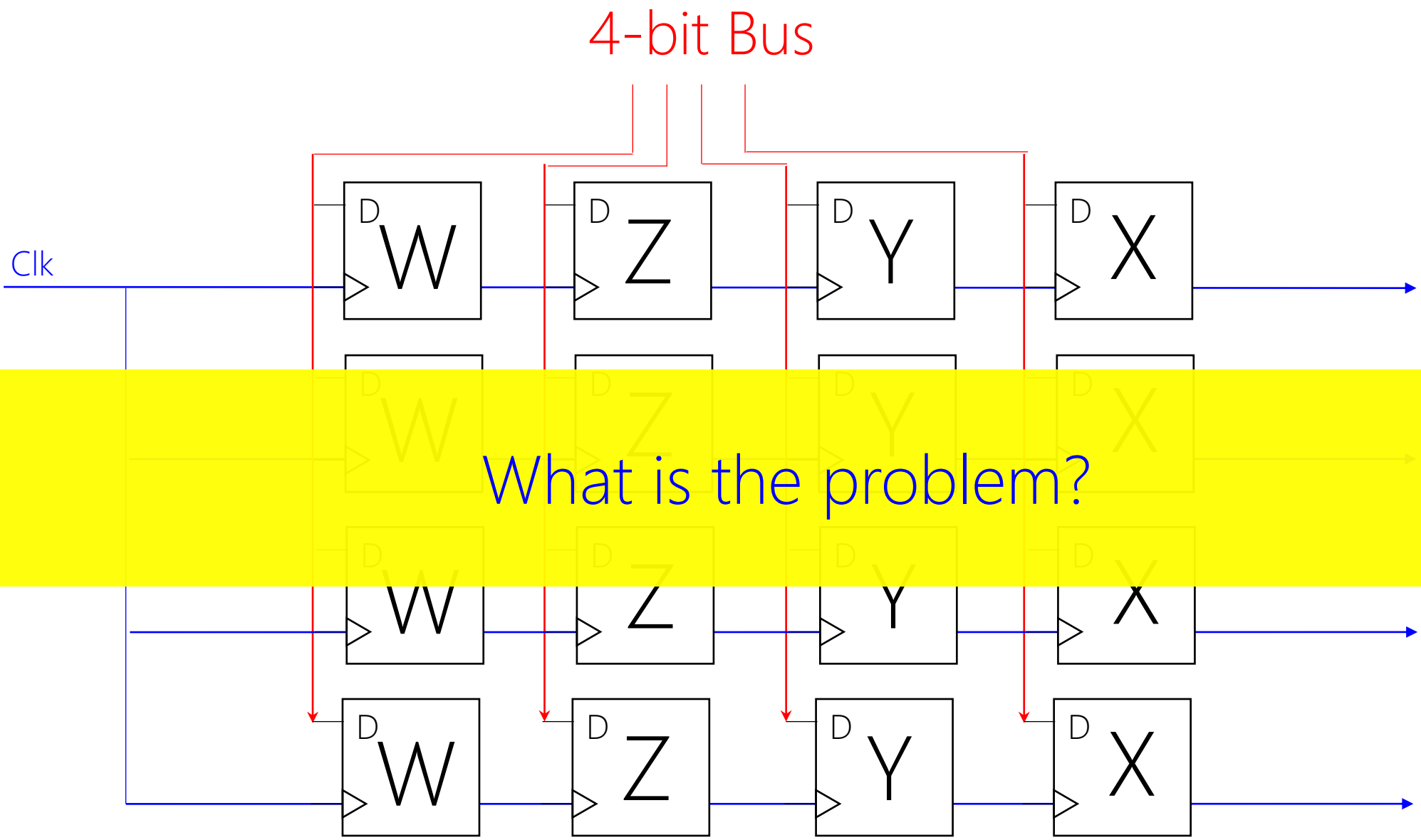
2 × Numbers ⟶ 2 × n FFs ⟶ 2 × n-bit Registers



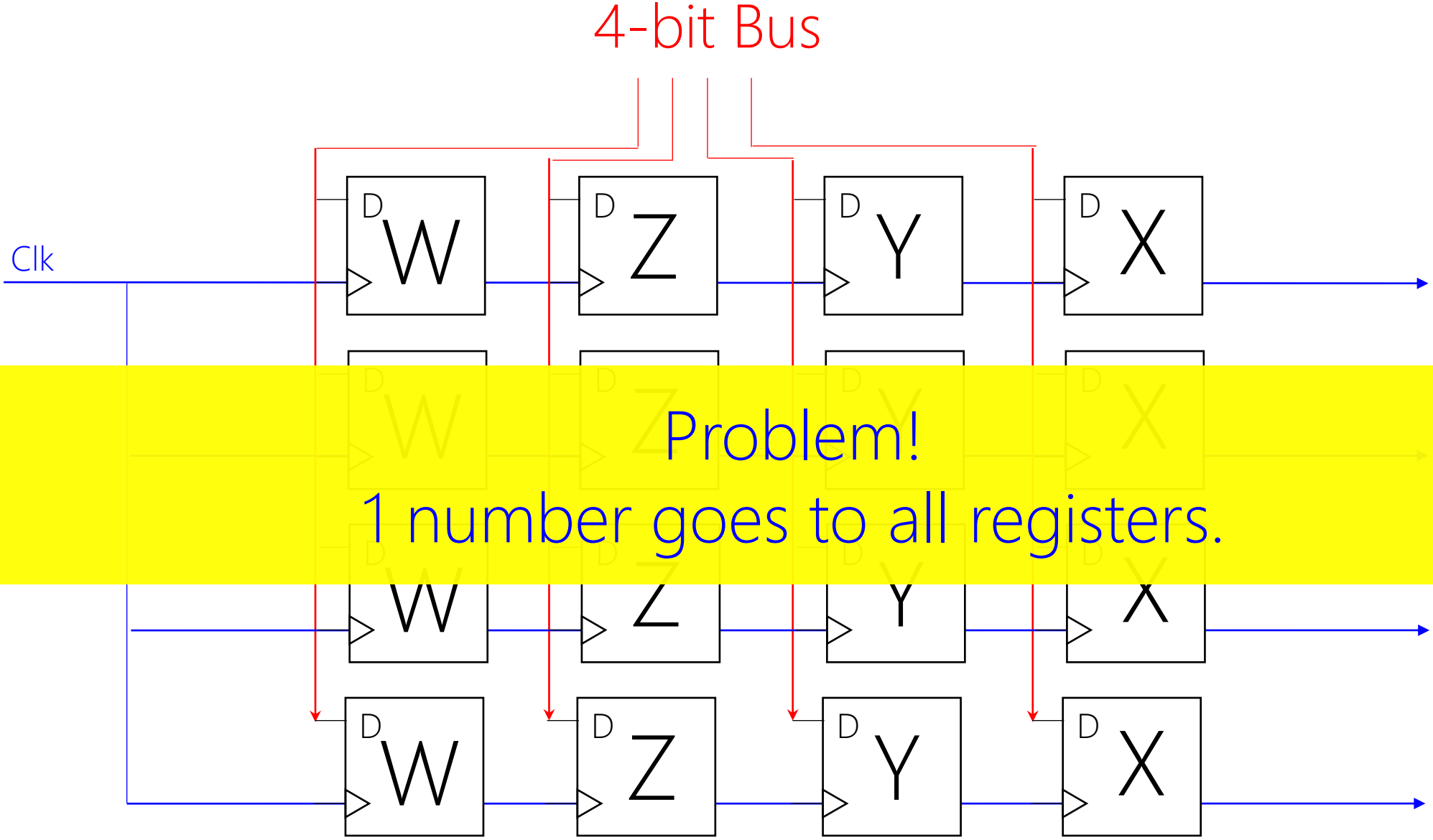
3 × Numbers ⟶ 3 × n FFs ⟶ 3 × n-bit Registers



4 × Numbers ⟶ 4 × n FFs ⟶ 4 × n-bit Registers



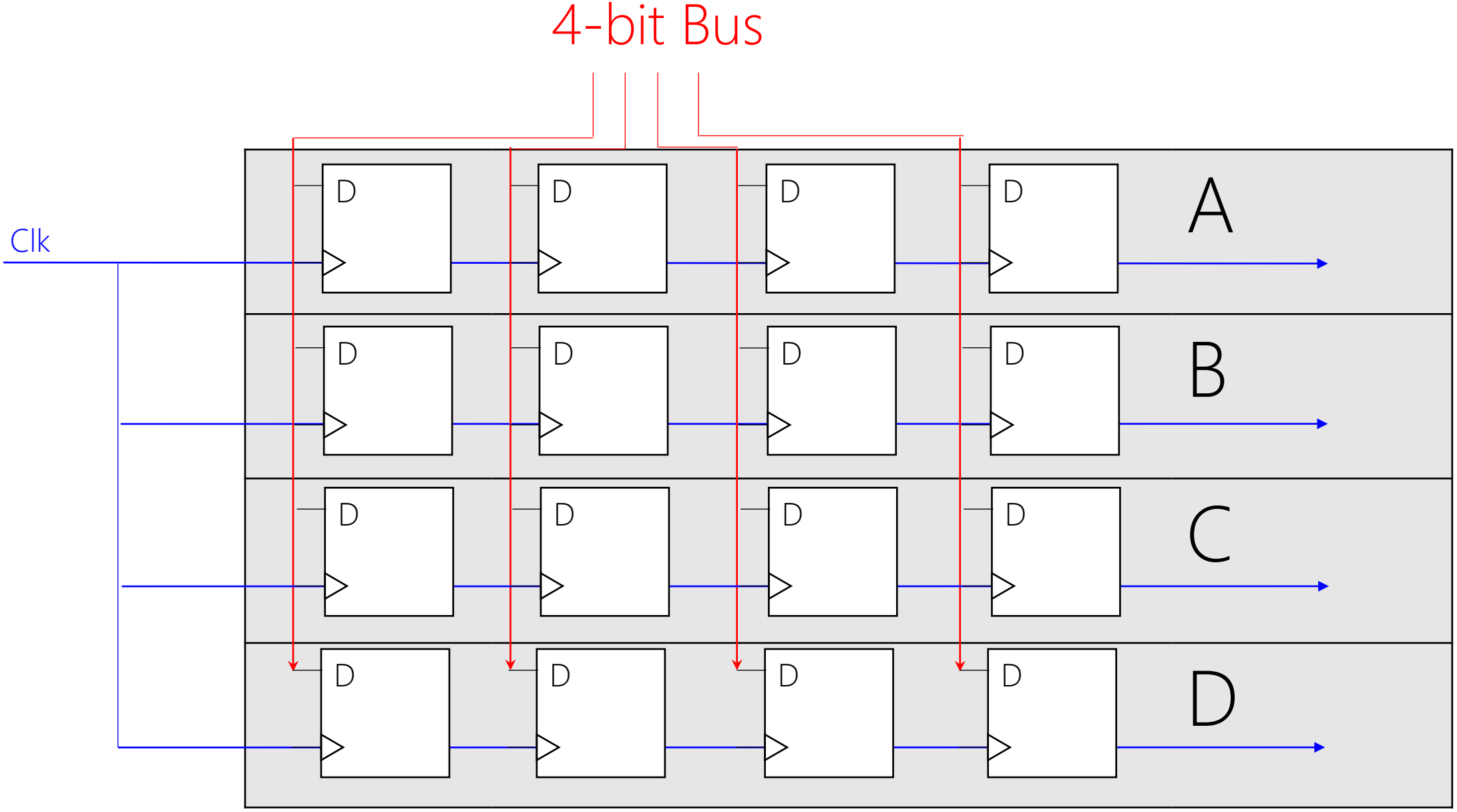
4 × Numbers ⟶ 4 × n FFs ⟶ 4 × n-bit Registers



4 × Numbers ⟶ 4 × n FFs ⟶ 4 × n-bit Registers

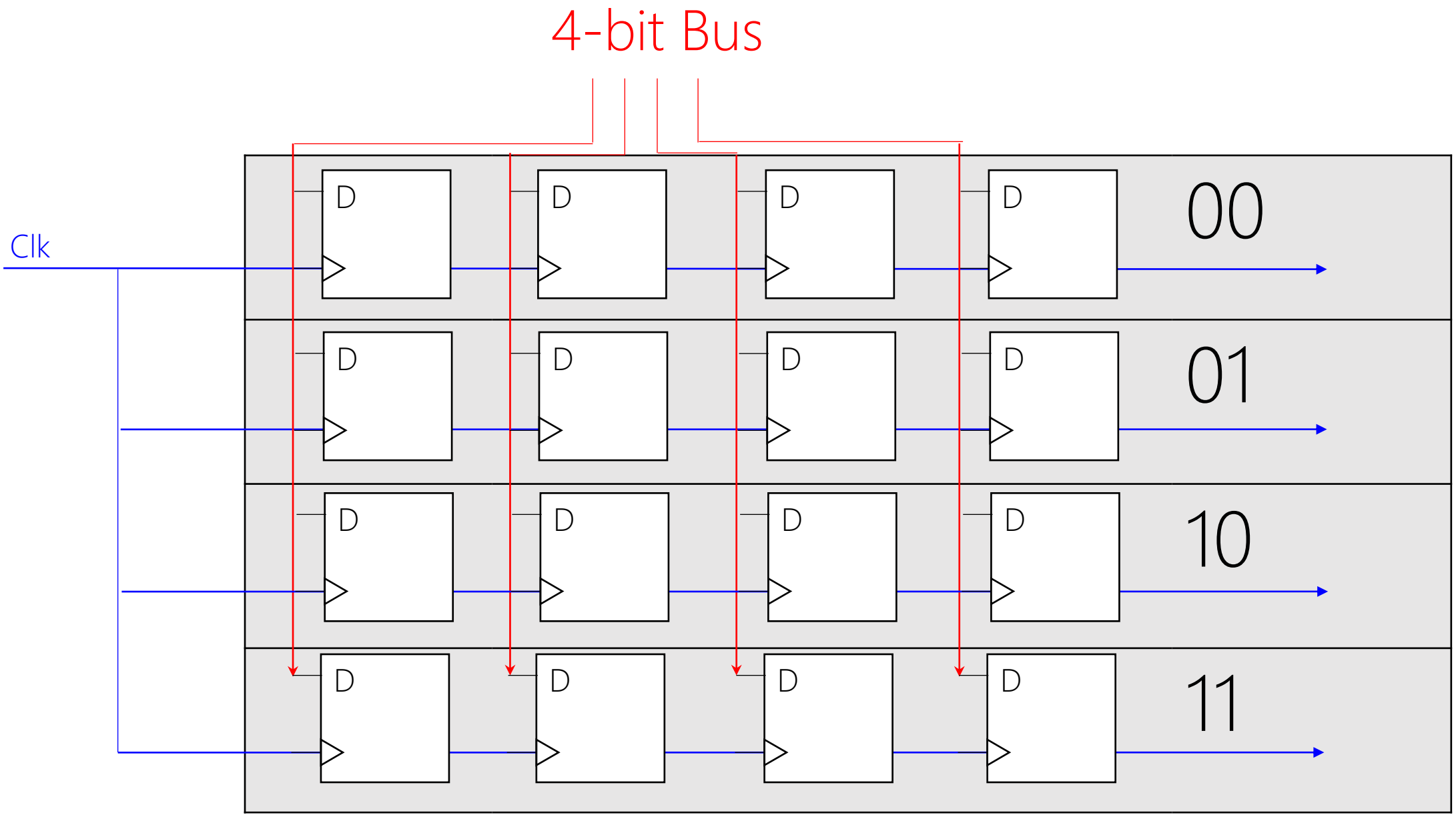
Which Register?

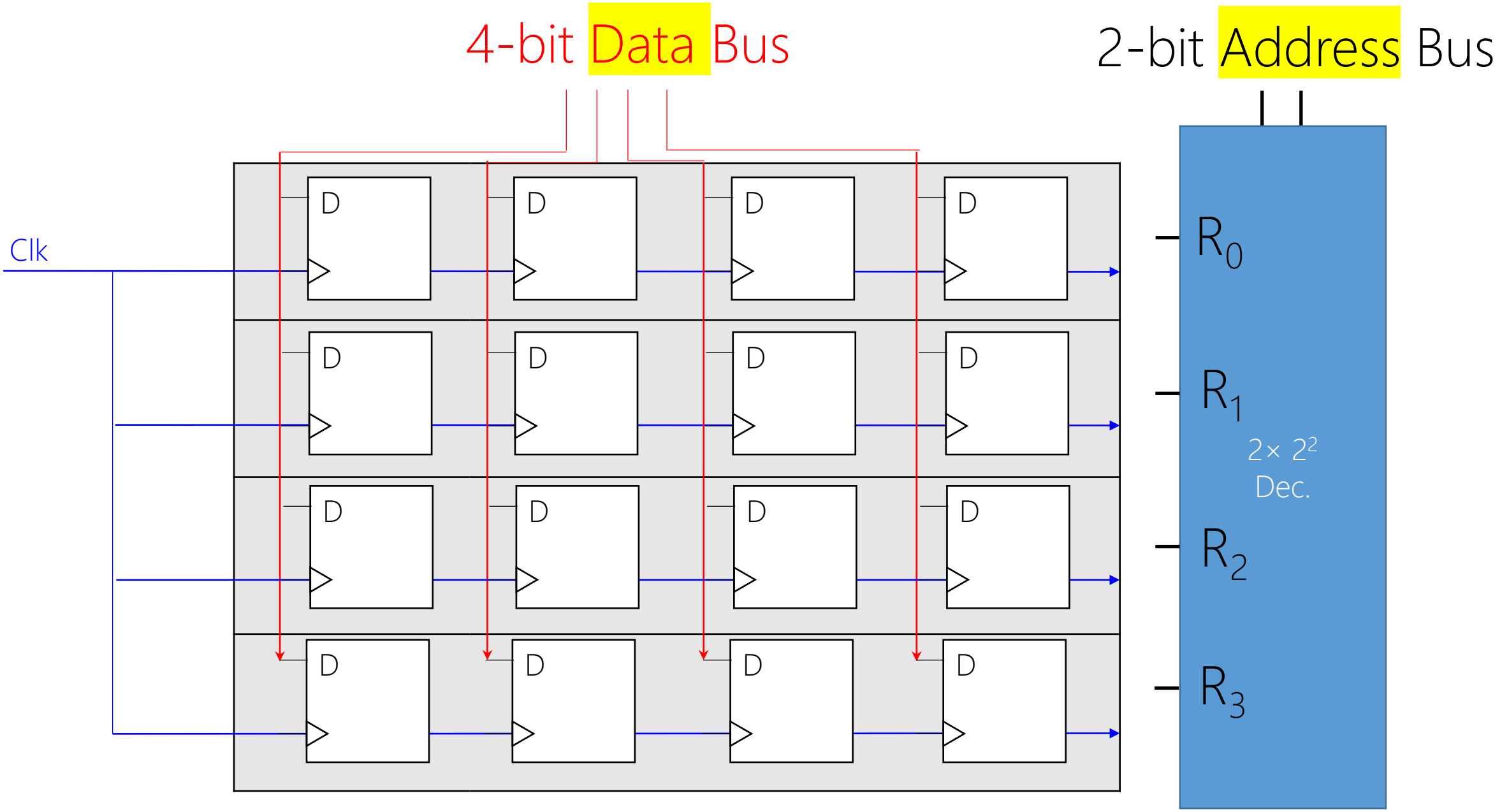
Select Register by Name?

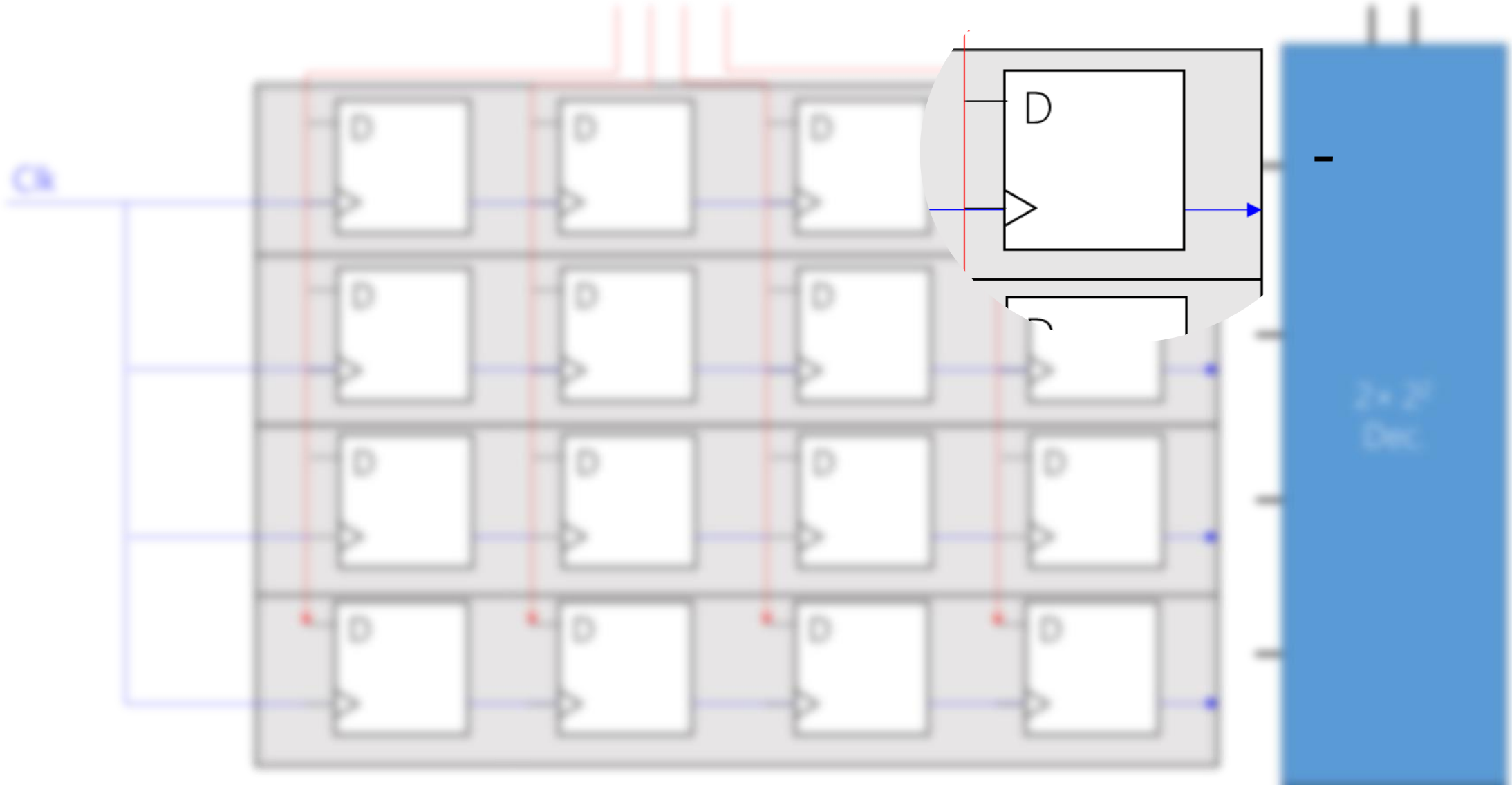


4 × Numbers ⟶ 4 × n FFs ⟶ 4 × n-bit Registers

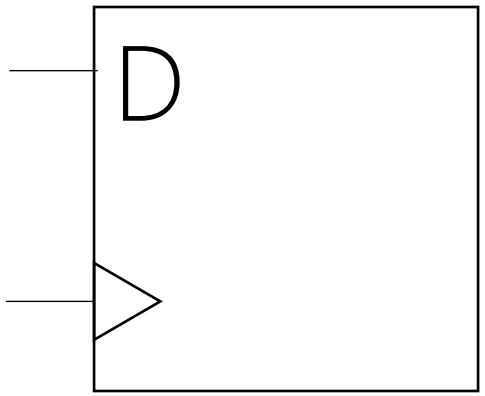
Select Register by Address?


4-bit Bus
Clk
D
00
01
10
11


4-bit Data Bus
2-bit Address Bus
Clk
D
R0
R1
R2
R3
2× 2²
Dec.


Clk
D
D
D
D
D
D
D
D
D
D
D
D
D
D
D
D

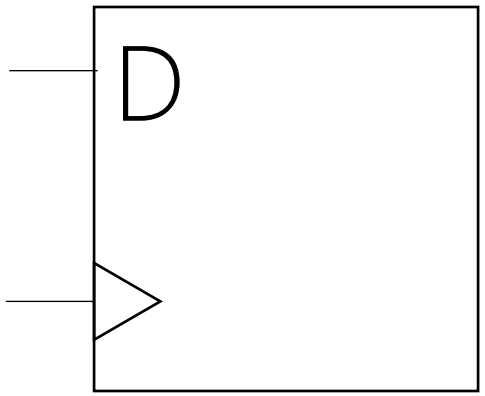
Input: I_0



Select: R_0

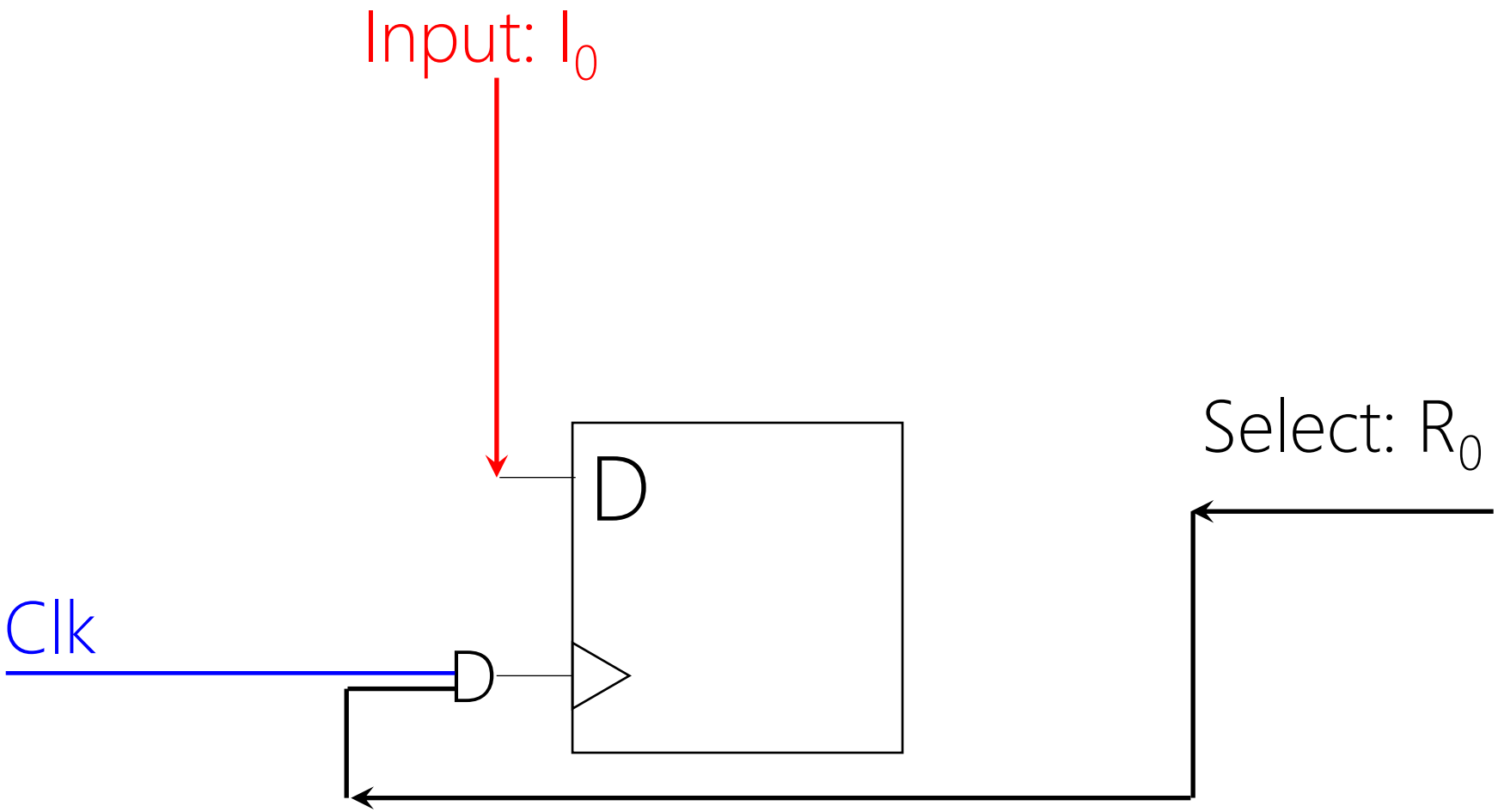
if R_0 (selected) then Write (Load) ($I_0 \longrightarrow$ D-FF)
else Store

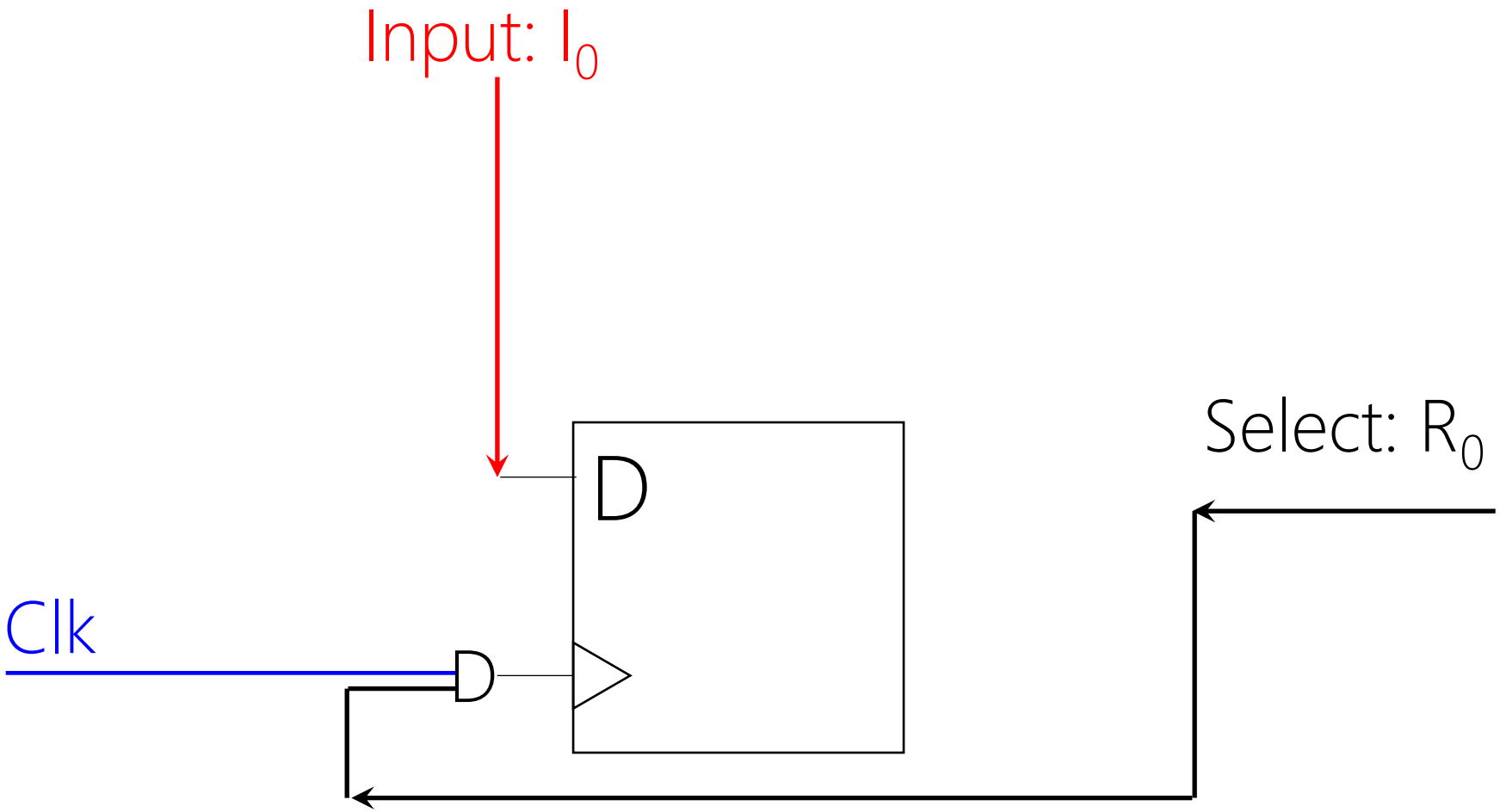
Input: I_0



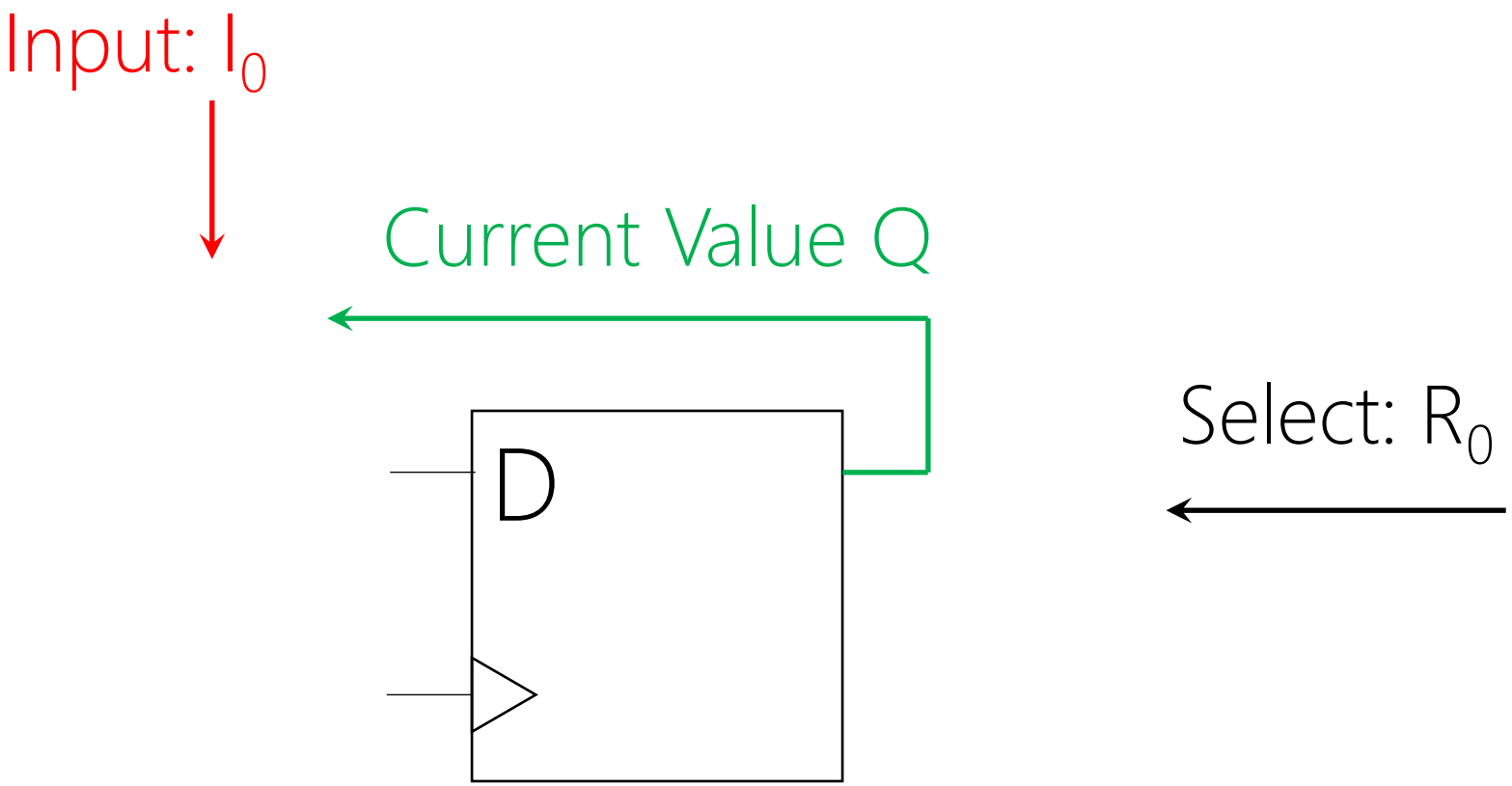
Select: R_0

if R_0 (selected) then Write (Load) ($I_0 \longrightarrow$ D-FF)
else Store but D-FF does not have store action!

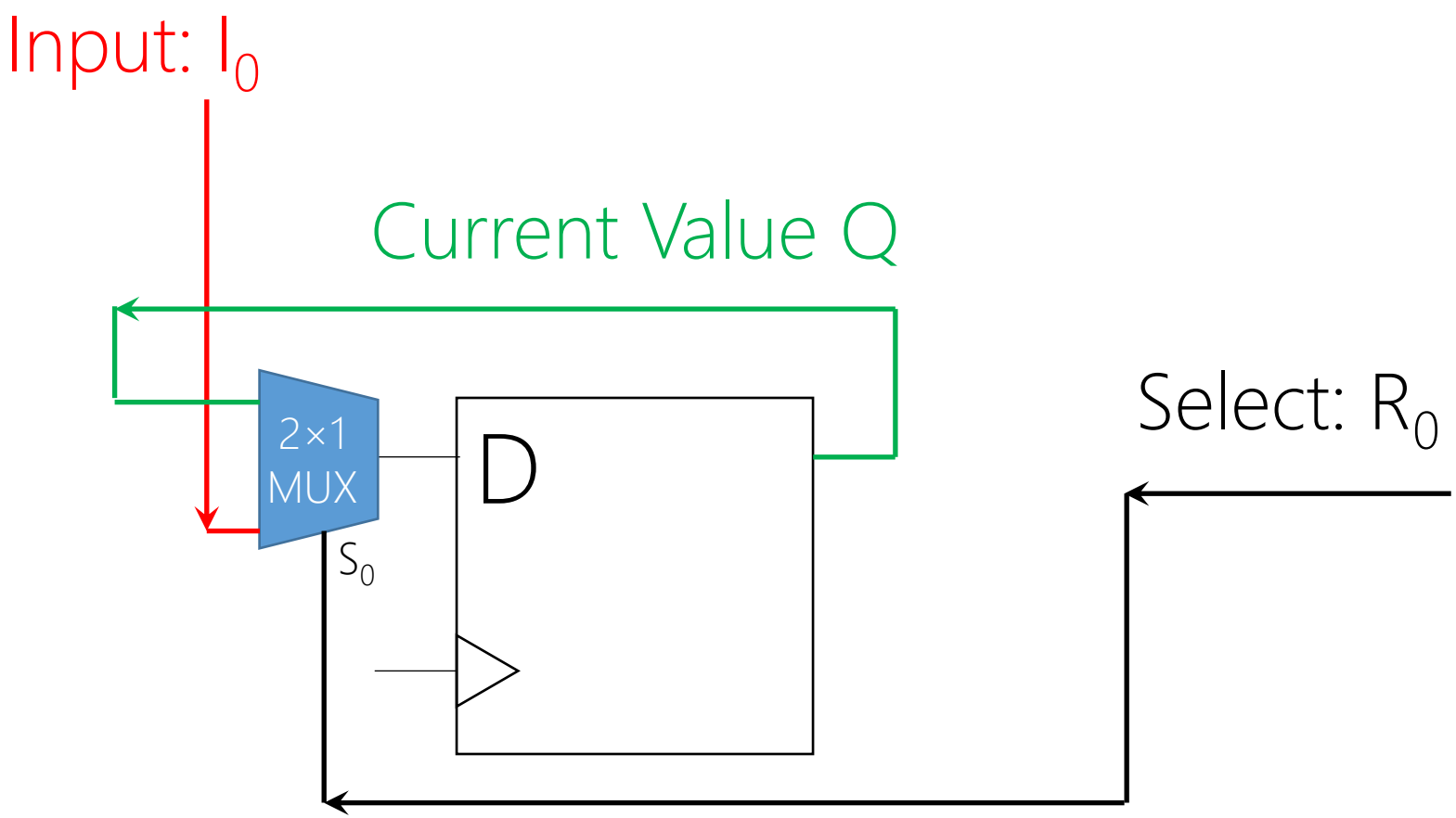

Input: I_0
D
Select: R_0
Clk
D



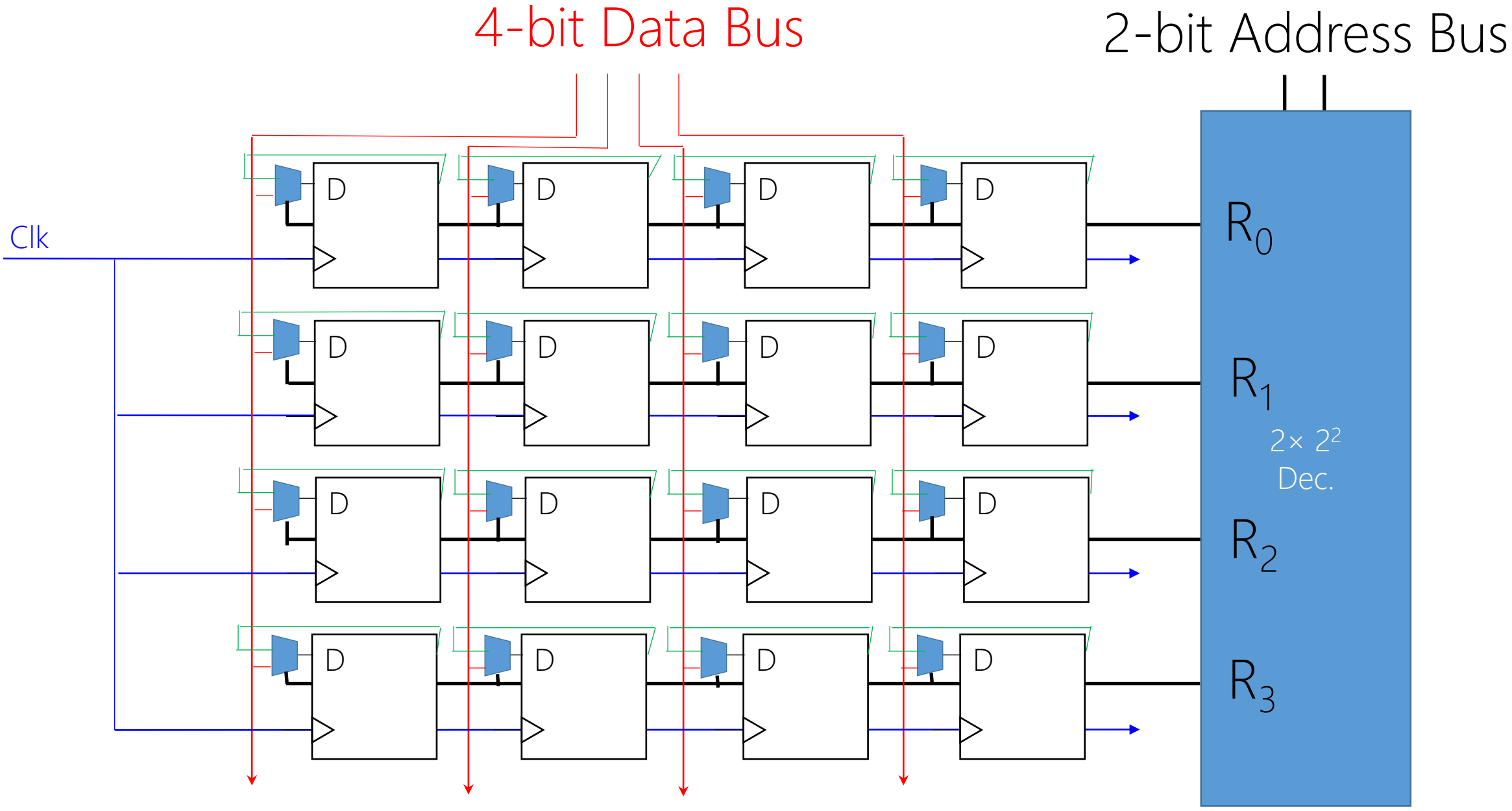
Never ever touch clock! Why?

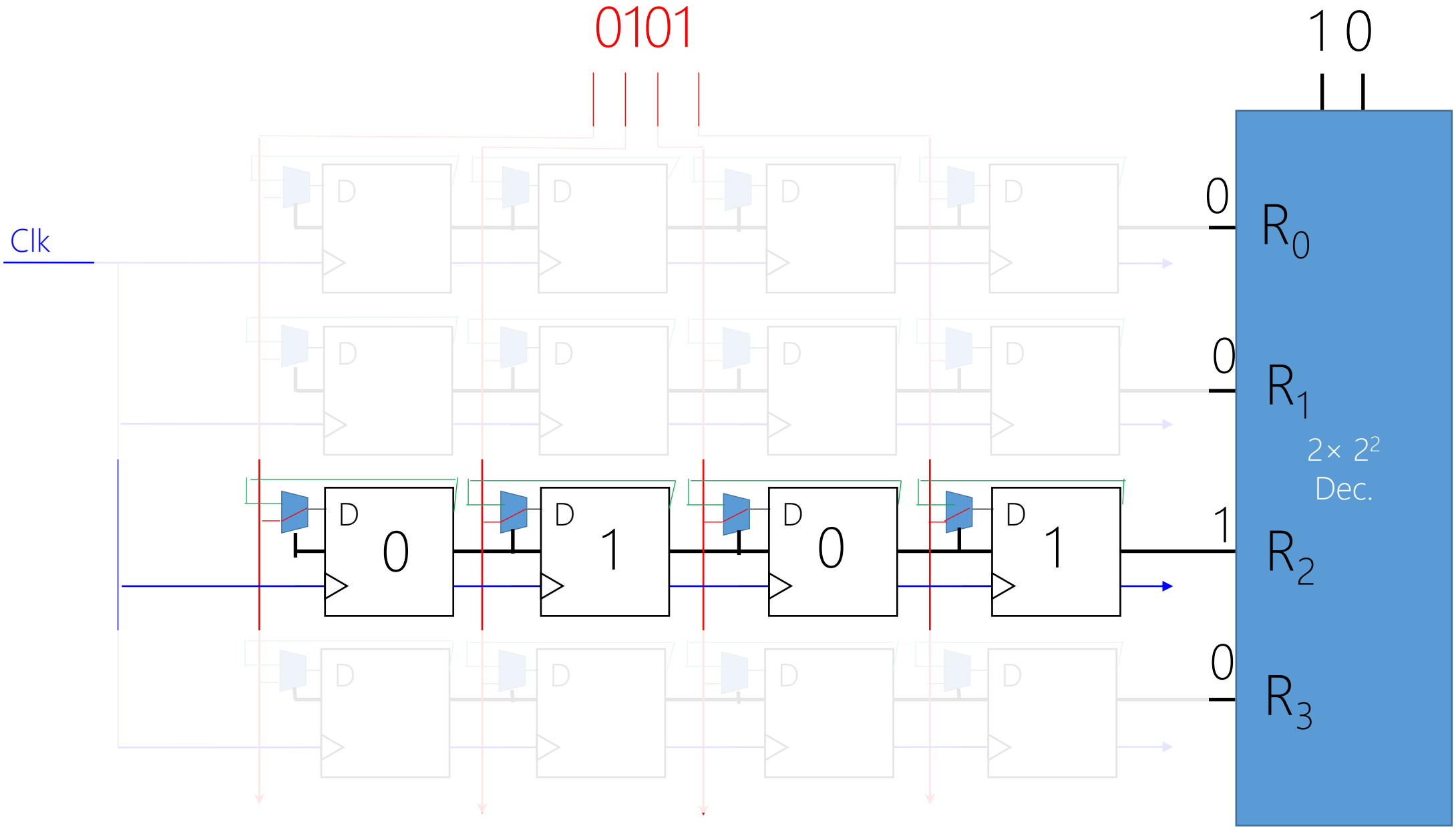


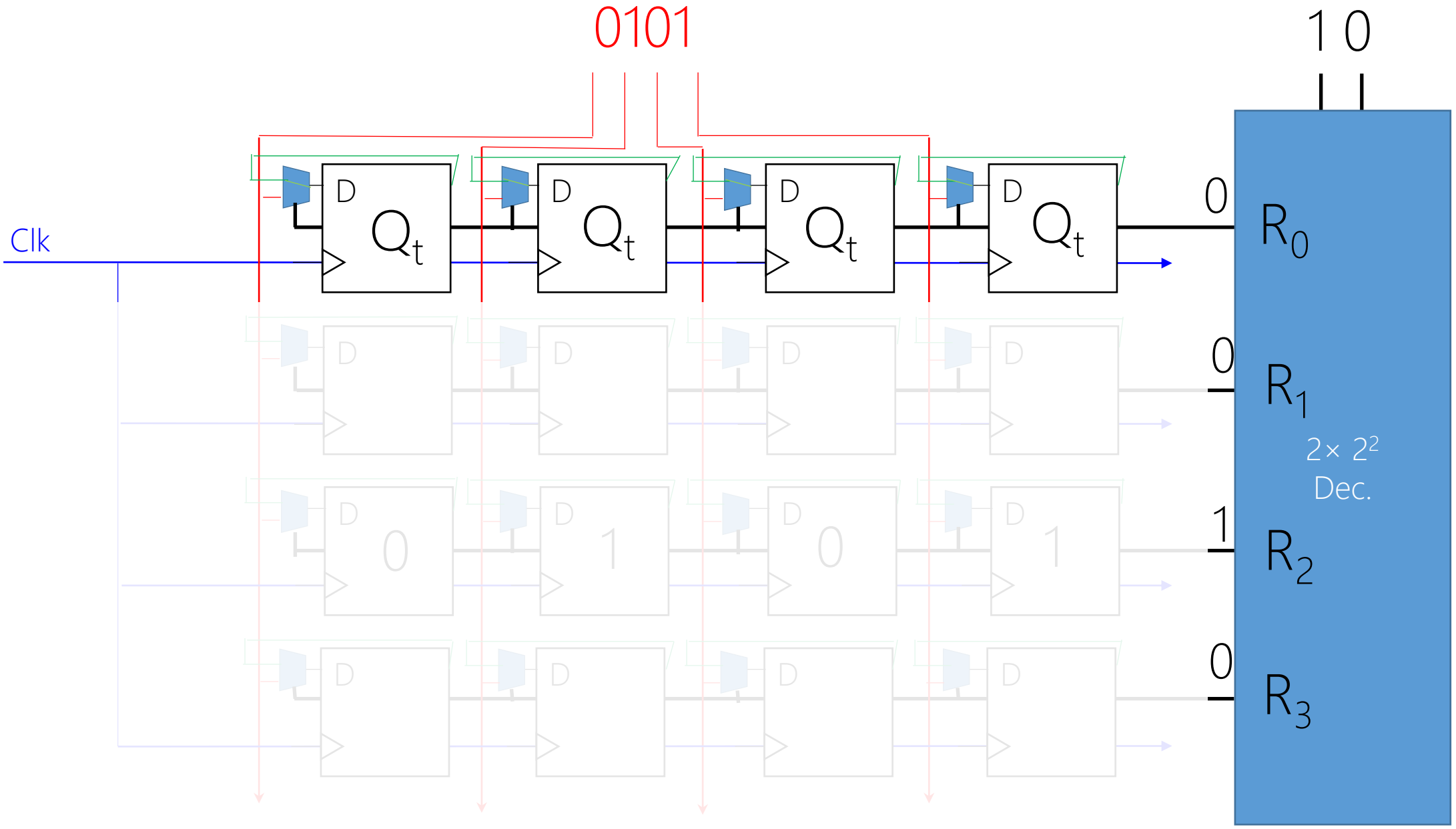
if R_0 (selected) then Write(Load) ($I_0 \rightarrow$ D-FF)
else Store

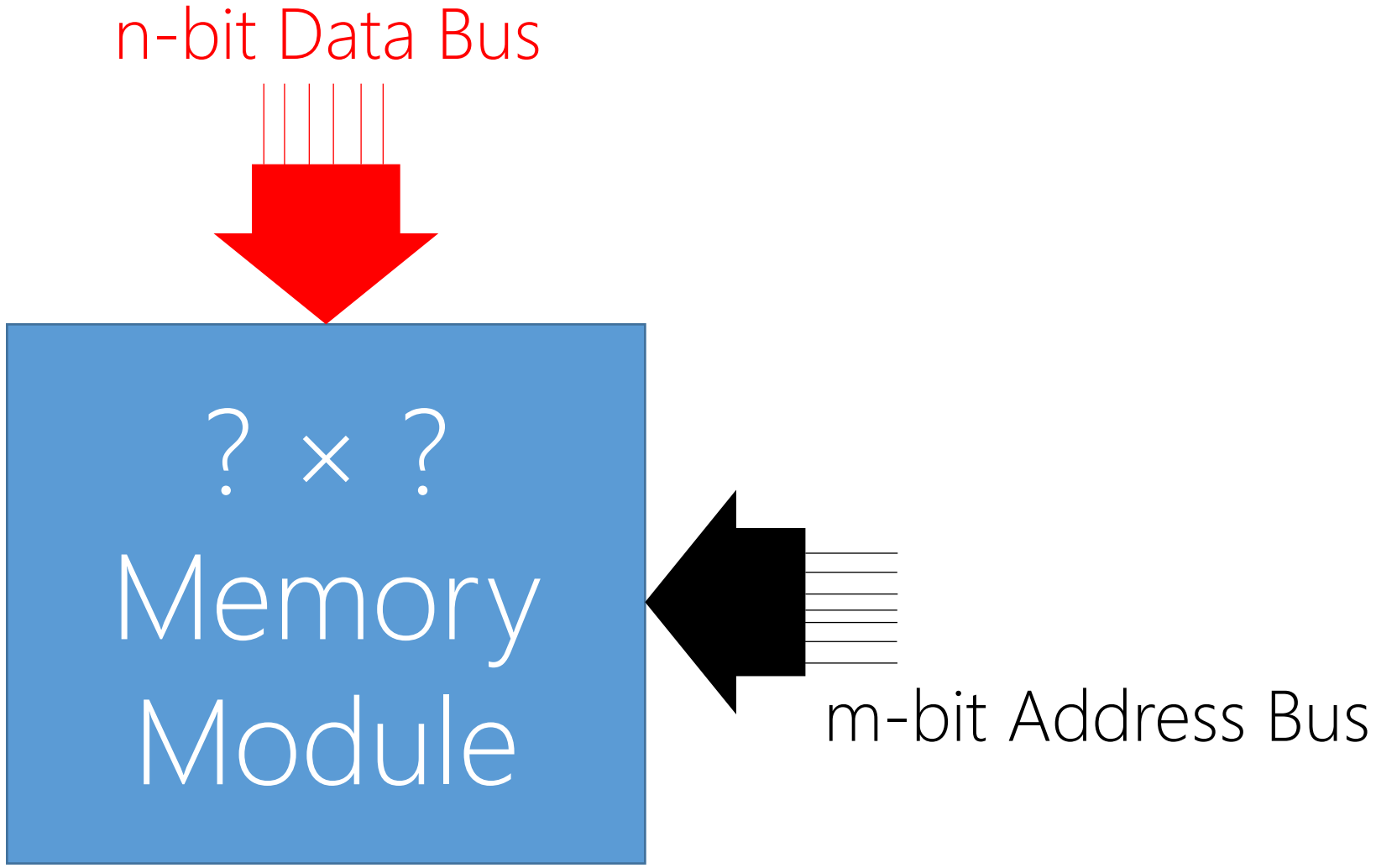


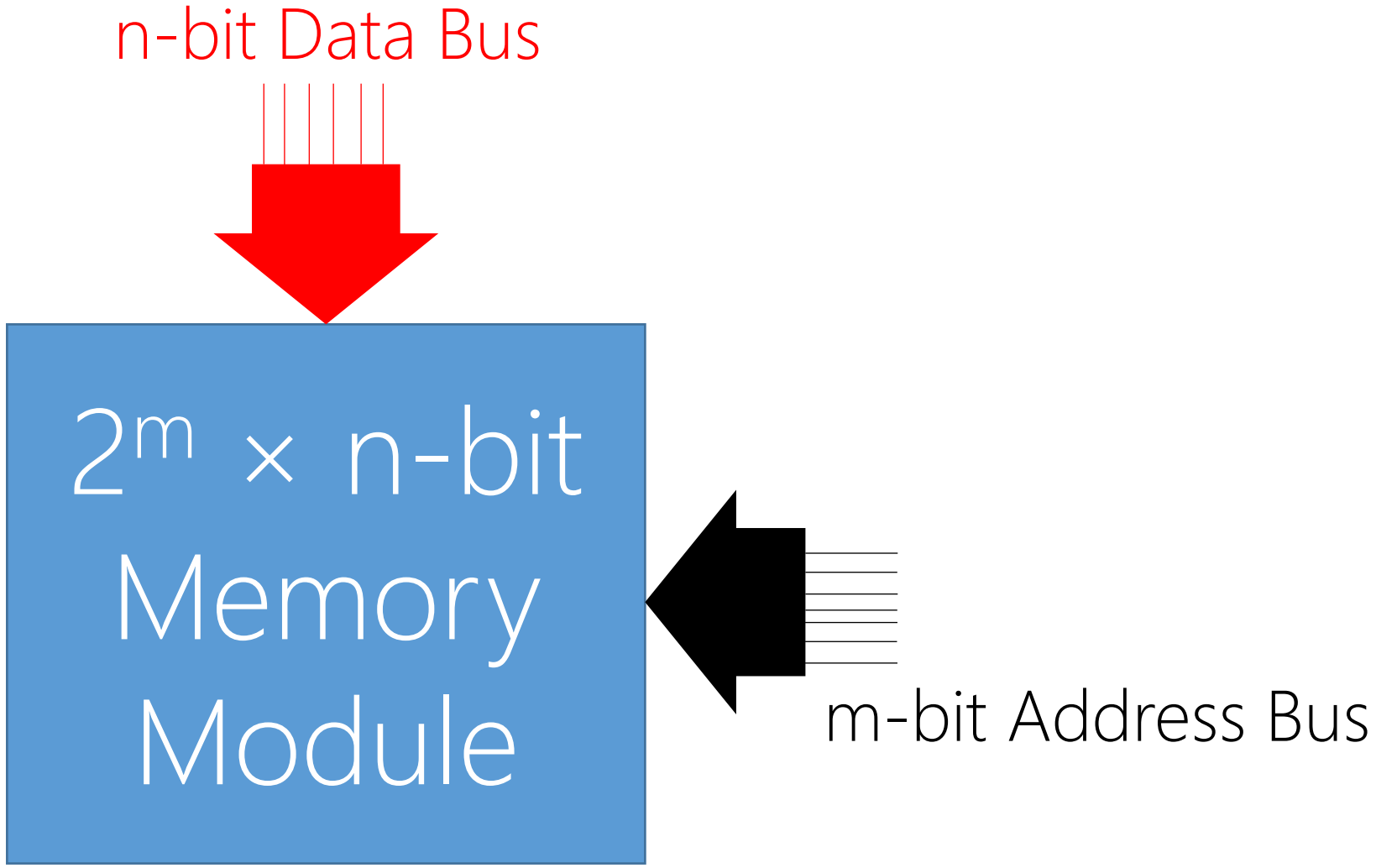
if R_0 (selected) then Write(Load) ($I_0 \rightarrow$ D-FF)
else Store

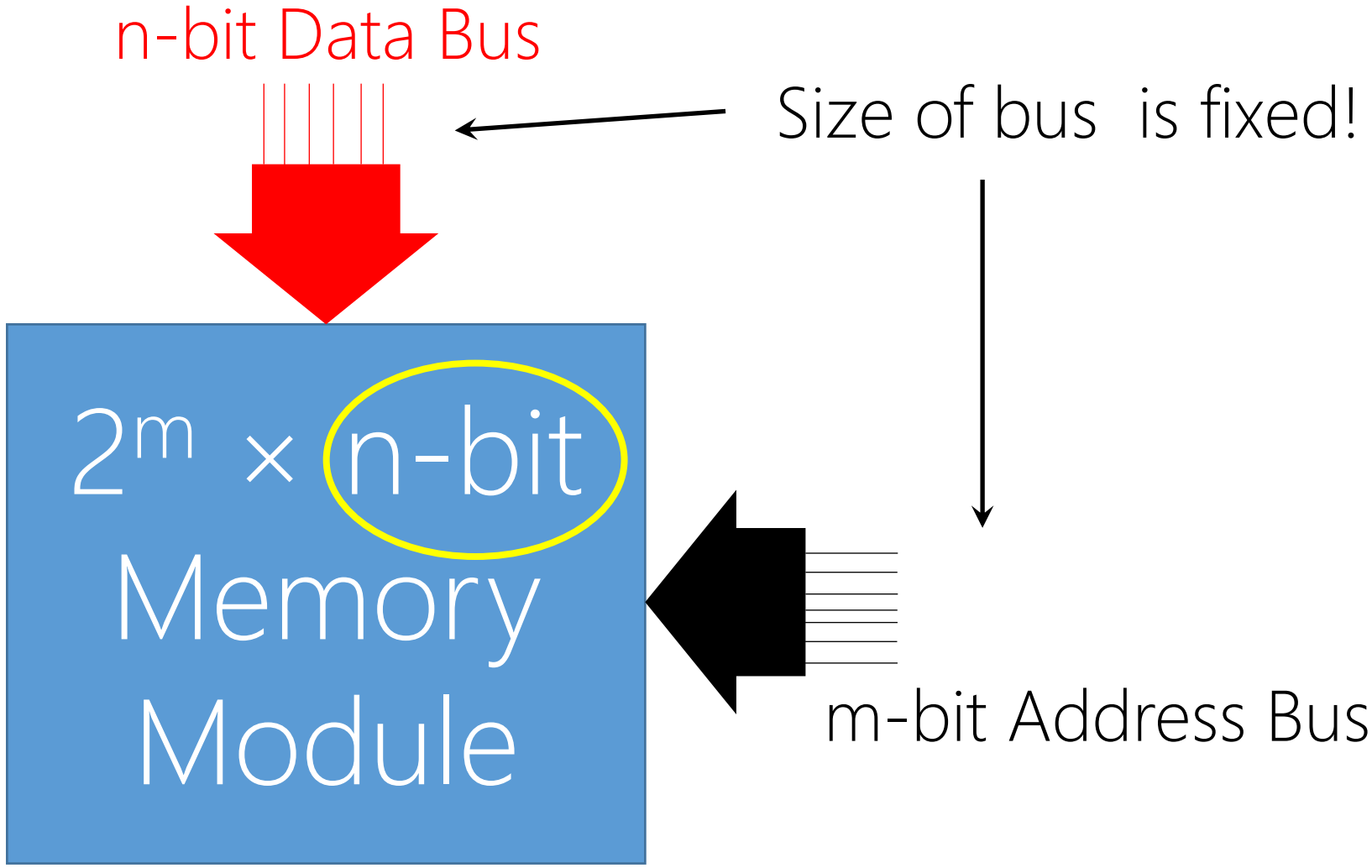

4-bit Data Bus
2-bit Address Bus
Clk
D
R0
R1
2× 2²
Dec.
R2
R3

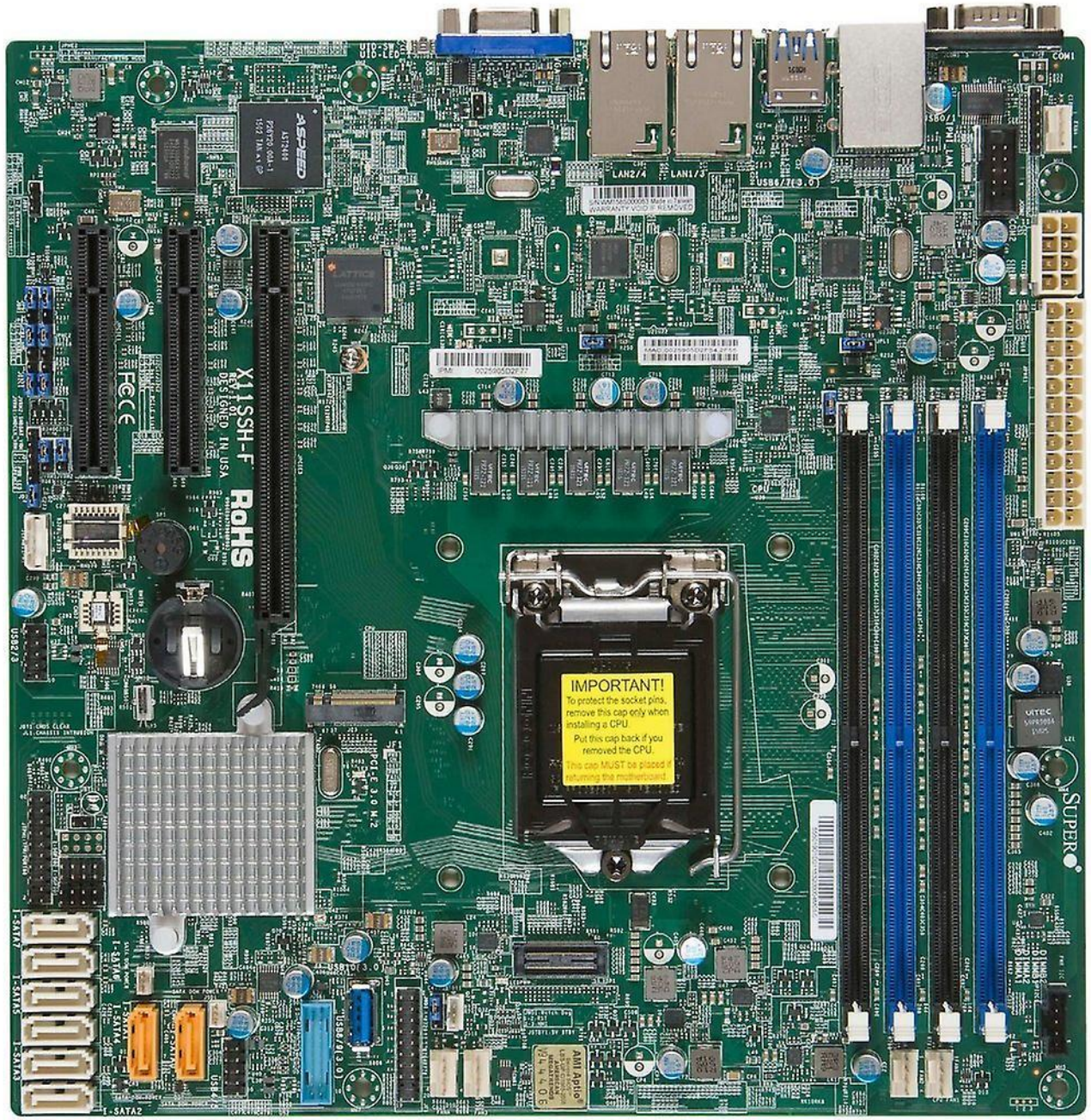

0101
1 0
Clk
D
0
D
1
D
0
D
1
0
R_0
0
R_1
2×2^2
Dec.
1
R_2
0
R_3


0101
1 0
Clk
D
Qt
0
R0
0
R1
2× 22
Dec.
1
R2
0
R3


n-bit Data Bus
? × ?
Memory
Module
m-bit Address Bus


n-bit Data Bus
$2^m \times$ n-bit Memory Module
m-bit Address Bus


n-bit Data Bus
Size of bus is fixed!
2m × n-bit
Memory
Module
m-bit Address Bus



64-bit Data Bus:

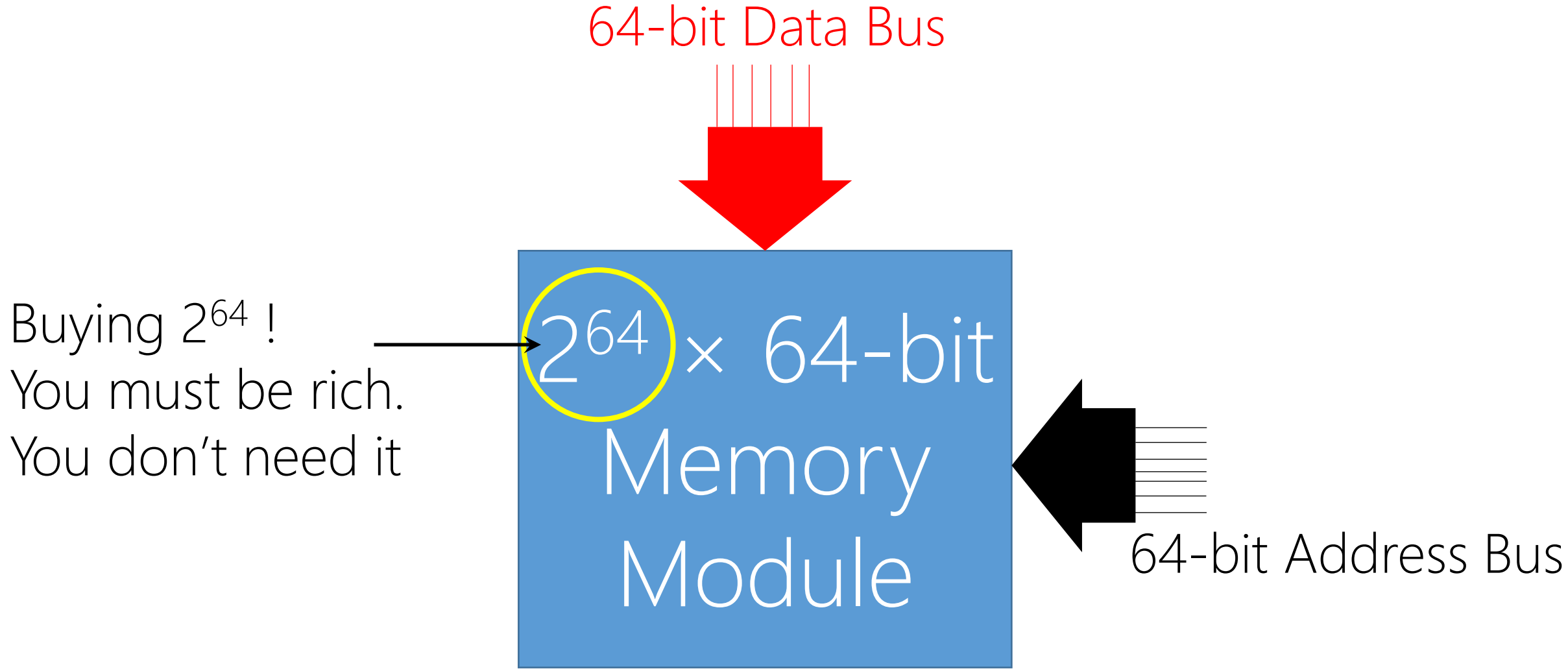
Numbers $\in [0,2^{64})$

Long Unsigned Integer in C/C++

How about signed integers?

64-bit Address Bus:

2^{64} Memory Locations



$2^{64}=2^{10}\times2^{10}\times2^{10}\times2^{10}\times2^{10}\times2^{10}\times2^{4}$ = 16 Exa Memory Locations

Kilo→Mega→Giga→Peta→Exa




Crucial RAM 16GB DDR4 2400 MHz CL17 Desktop Memory CT16G4DFD824A

Visit the Crucial Store

9,324 ratings | 46 answered questions

Price: CDN$ 81.01 & FREE Shipping

Get a **$5 promotional credit** on purchase of **$25 or more** in Amazon.ca Gift Cards. Restrictions Apply

$16 G = 16 \times 2^{10}$ (M) $= 16 \times 2^{10} \times 2^{10}$ (K) $= 16 \times 2^{10} \times 2^{10} \times 2^{10}$ Memory Locations

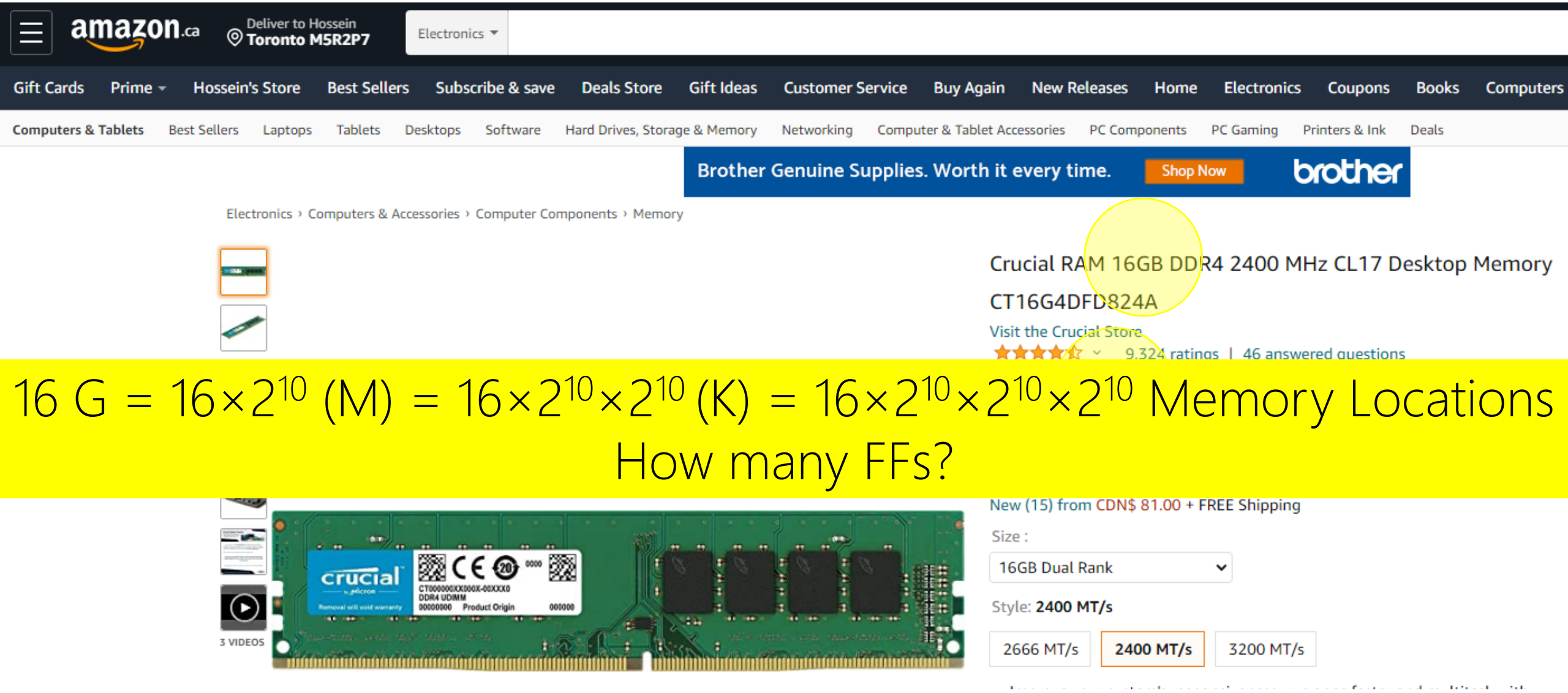


3 VIDEOS

16GB Dual Rank

Style: **2400 MT/s**

2666 MT/s | **2400 MT/s** | 3200 MT/s


amazon.ca
Deliver to Hossein
Toronto M5R2P7
Electronics
Gift Cards
Prime
Hossein's Store
Best Sellers
Subscribe & save
Deals Store
Gift Ideas
Customer Service
Buy Again
New Releases
Home
Electronics
Coupons
Books
Computers
Computers & Tablets
Best Sellers
Laptops
Tablets
Desktops
Software
Hard Drives, Storage & Memory
Networking
Computer & Tablet Accessories
PC Components
PC Gaming
Printers & Ink
Deals
Brother Genuine Supplies. Worth it every time.
Shop Now
brother
Electronics › Computers & Accessories › Computer Components › Memory
Crucial RAM 16GB DDR4 2400 MHz CL17 Desktop Memory
CT16G4DFD824A
Visit the Crucial Store
9,324 ratings | 46 answered questions
16 G = 16×2^10 (M) = 16×2^10×2^10 (K) = 16×2^10×2^10×2^10 Memory Locations
How many FFs?
3 VIDEOS
crucial
New (15) from CDN$ 81.00 + FREE Shipping
Size :
16GB Dual Rank
Style: 2400 MT/s
2666 MT/s
2400 MT/s
3200 MT/s

Crucial RAM 16GB DDR4 2400 MHz CL17 Desktop Memory

$16\,G = 16\times2^{10}\,(M) = 16\times2^{10}\times2^{10}\,(K) = 16\times2^{10}\times2^{10}\times2^{10}$ Memory Locations

How many FFs?

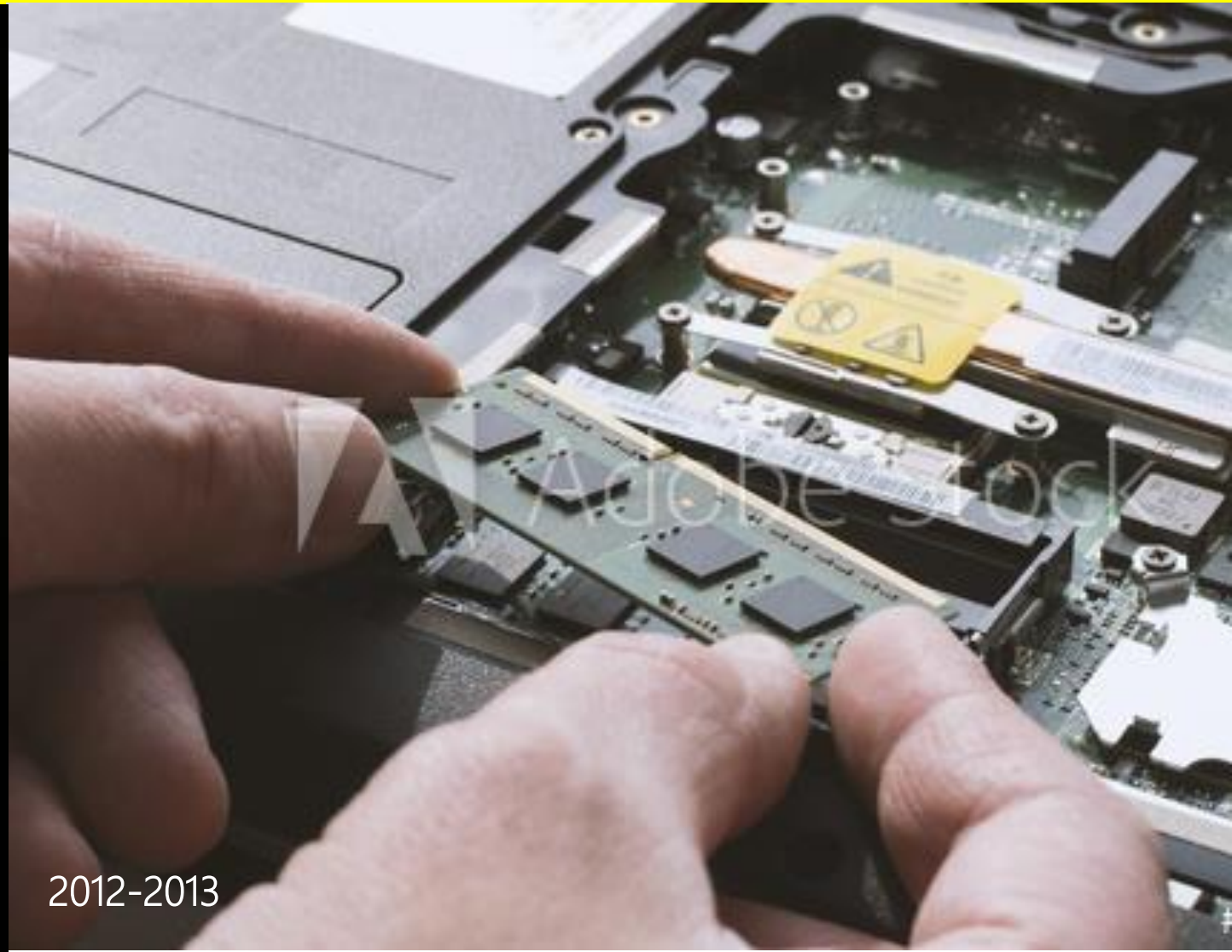
$(2^4=16)\times2^{10}\times2^{10}\times2^{10}\times(2^6=64\text{-bit Data Bus}) = 2^{40}$

$16\ G = 16 \times 2^{10}\ (M) = 16 \times 2^{10} \times 2^{10}\ (K) = 16 \times 2^{10} \times 2^{10} \times 2^{10}$ Memory Locations

How many FFs?

$(2^4 = 16) \times 2^{10} \times 2^{10} \times 2^{10} \times (2^6 = 64\text{-bit Data Bus}) = 2^{40}$

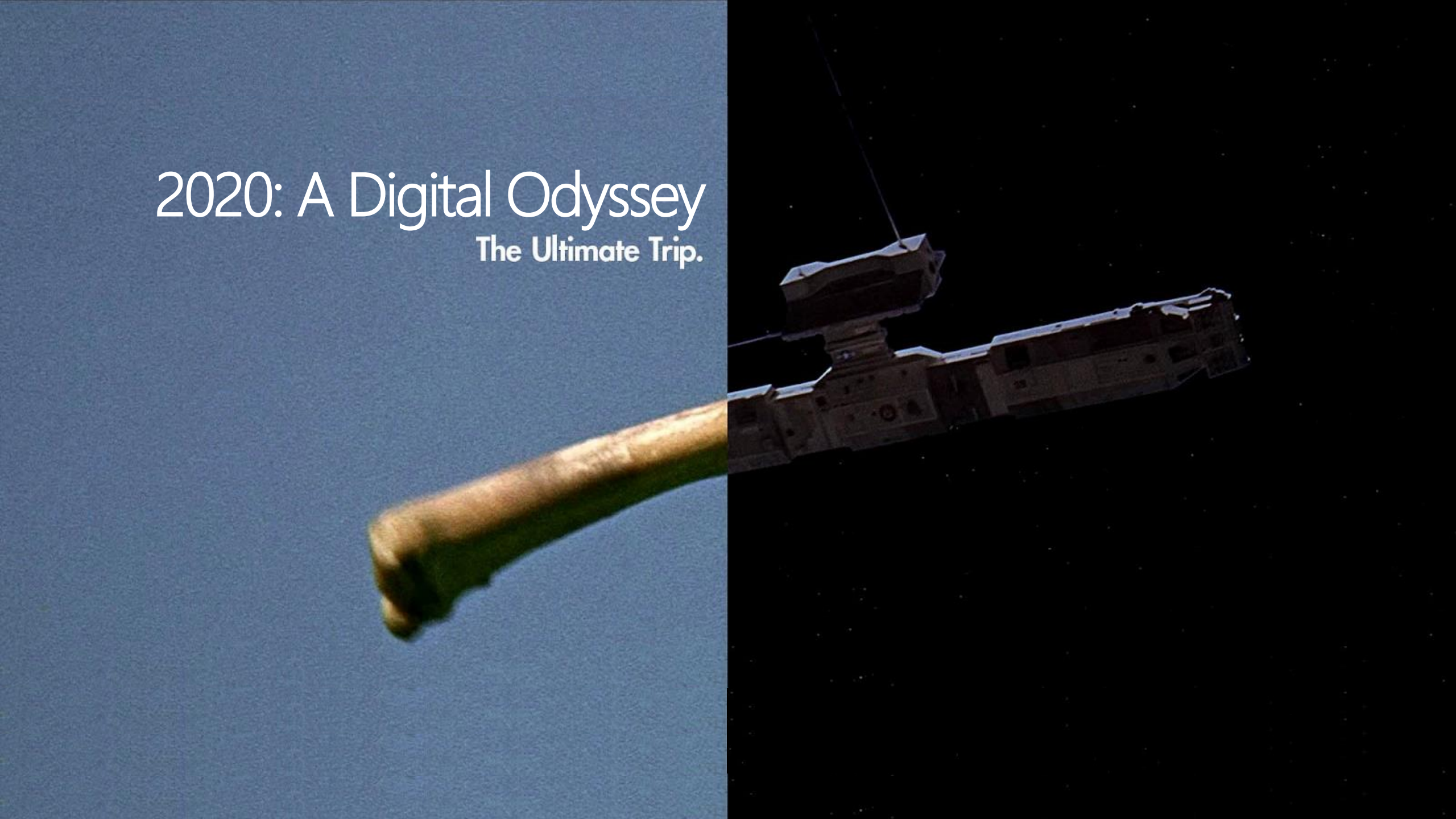
1958–1959

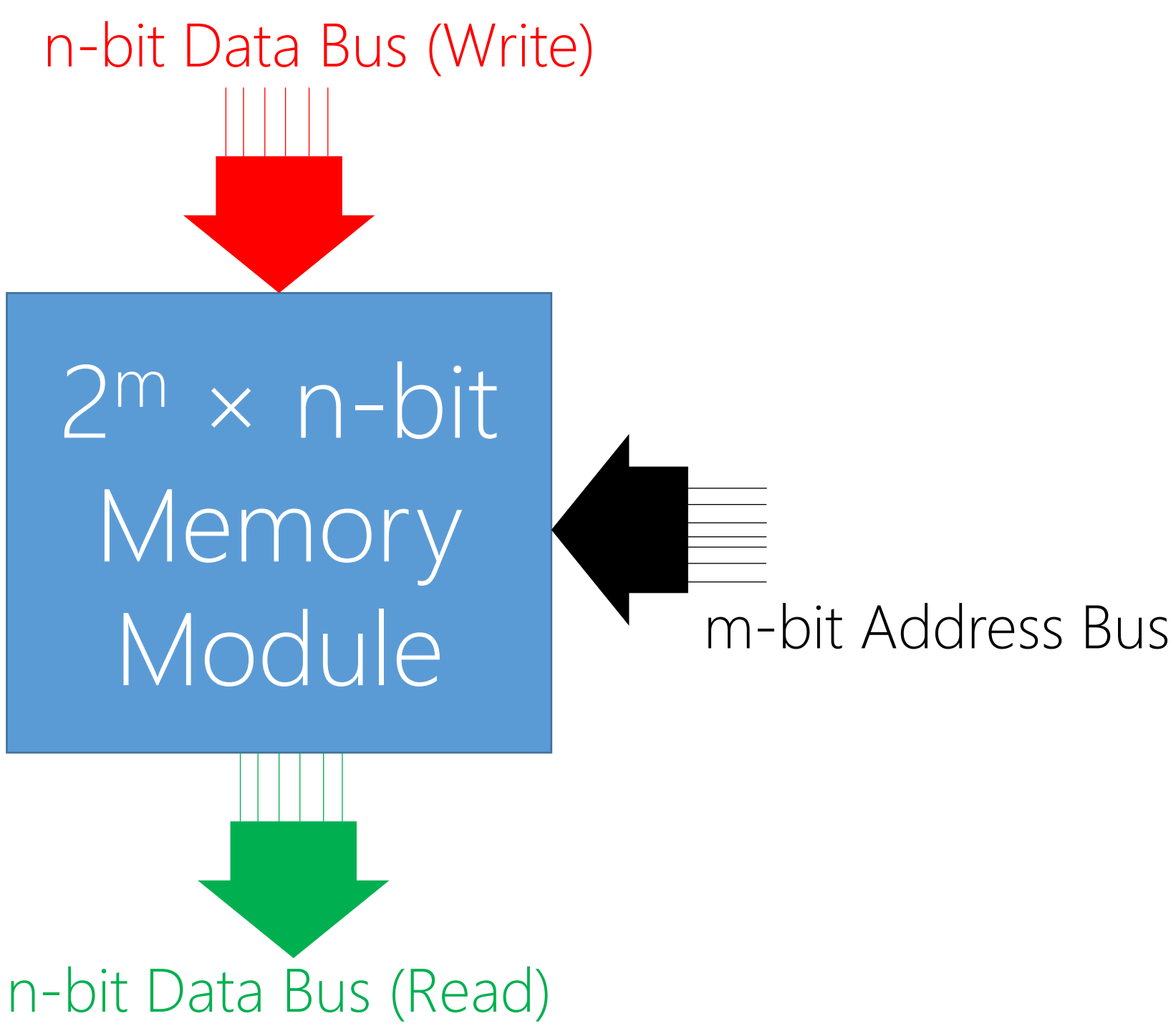


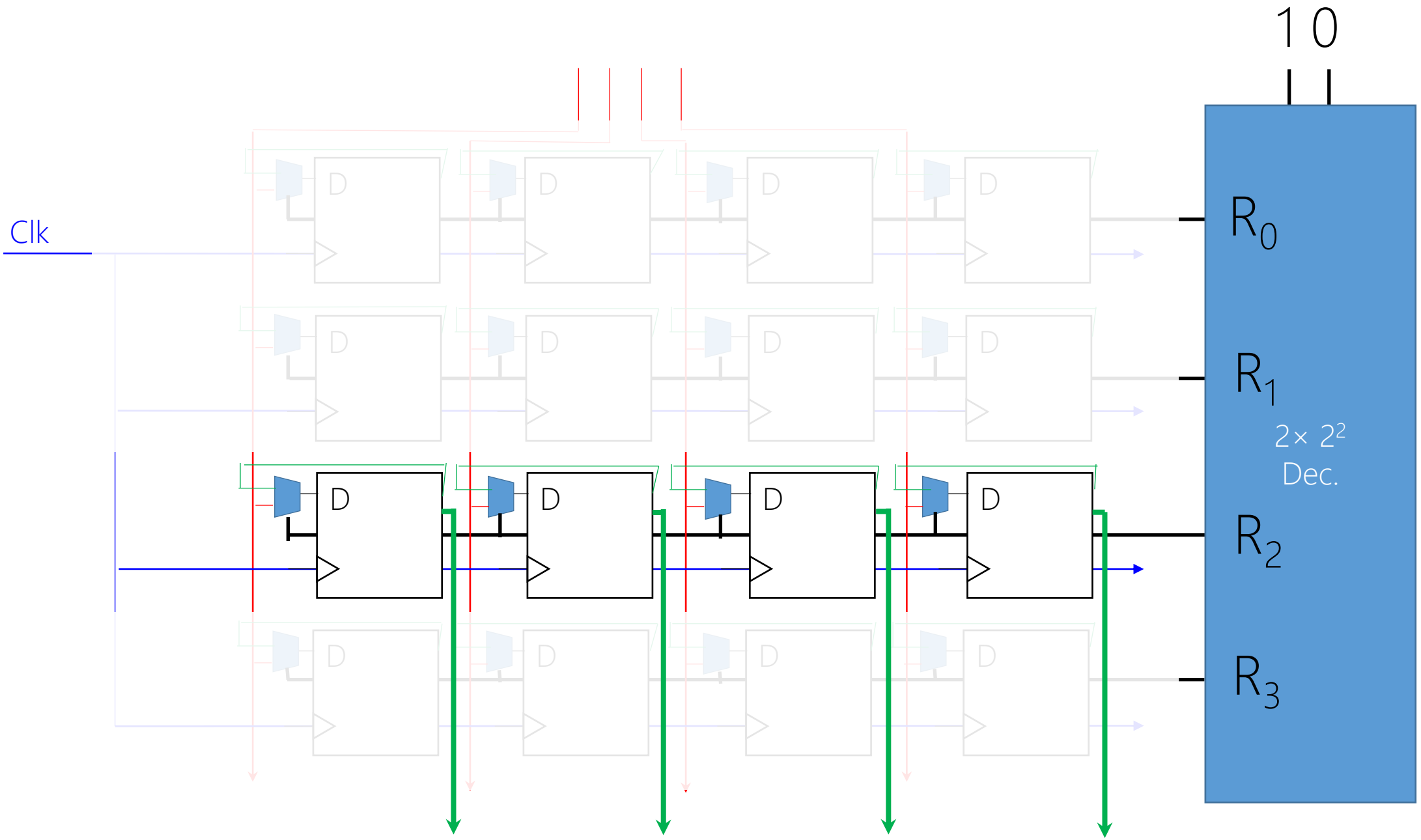
2012-2013

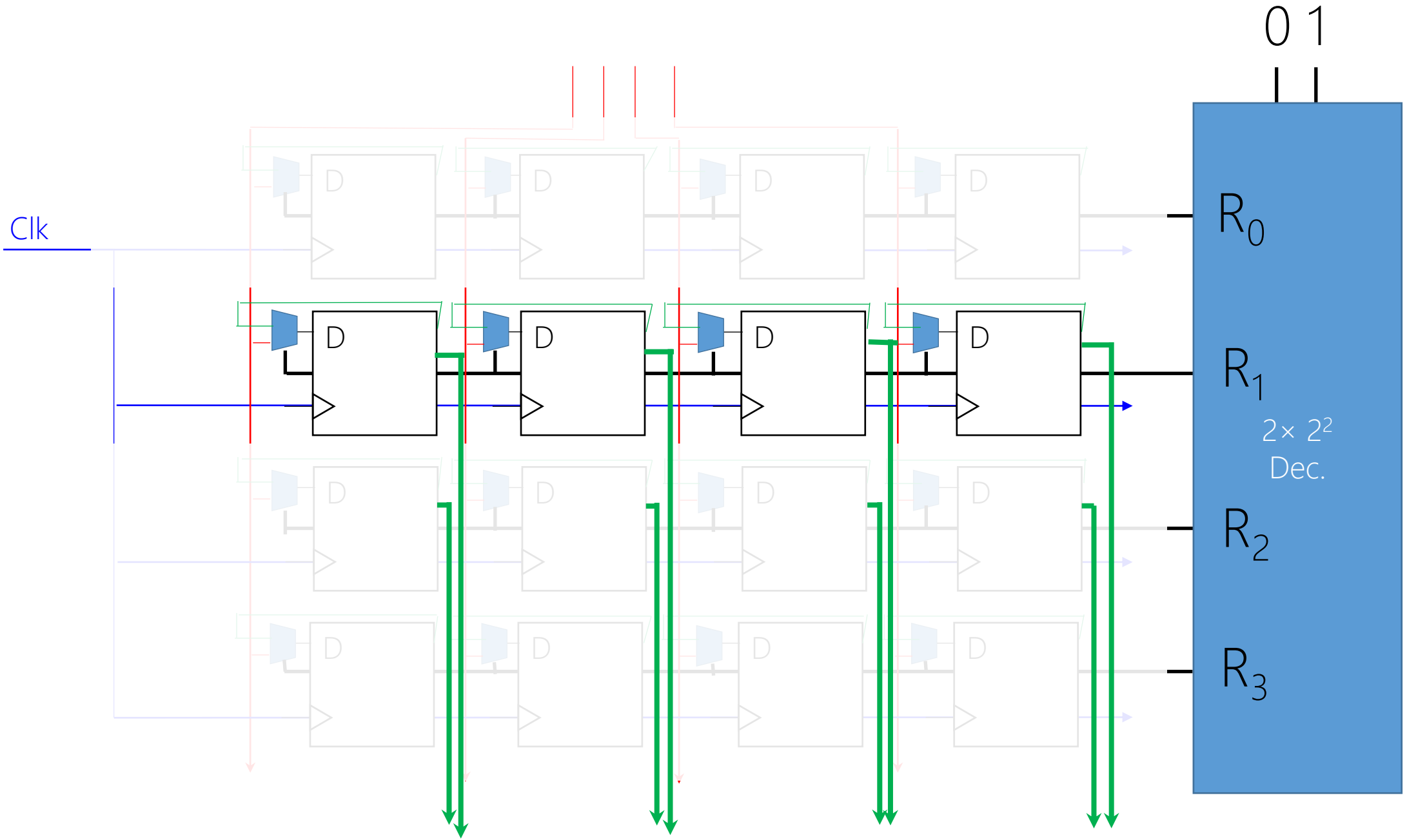
2020: A Digital Odyssey

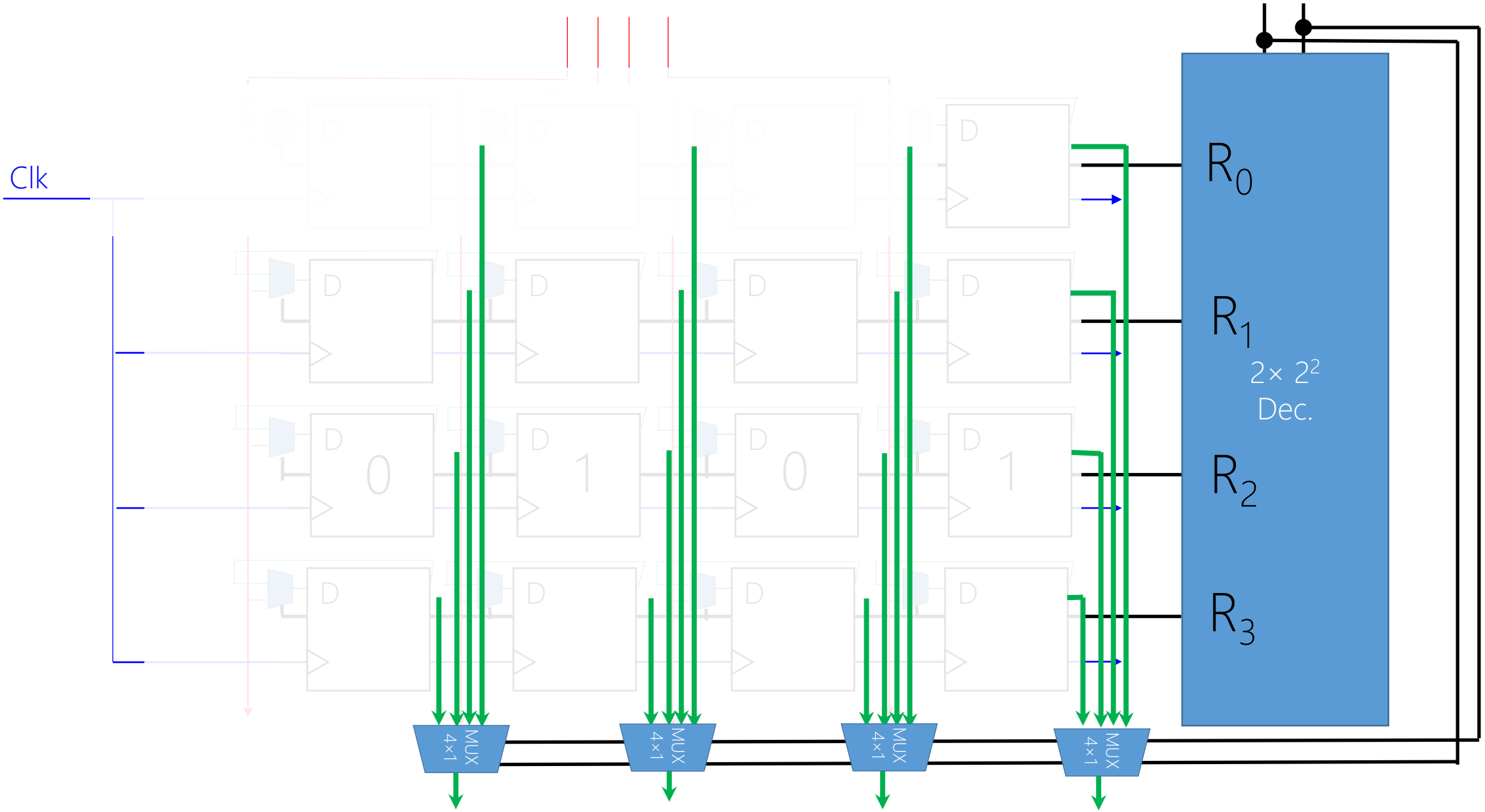
The Ultimate Trip.

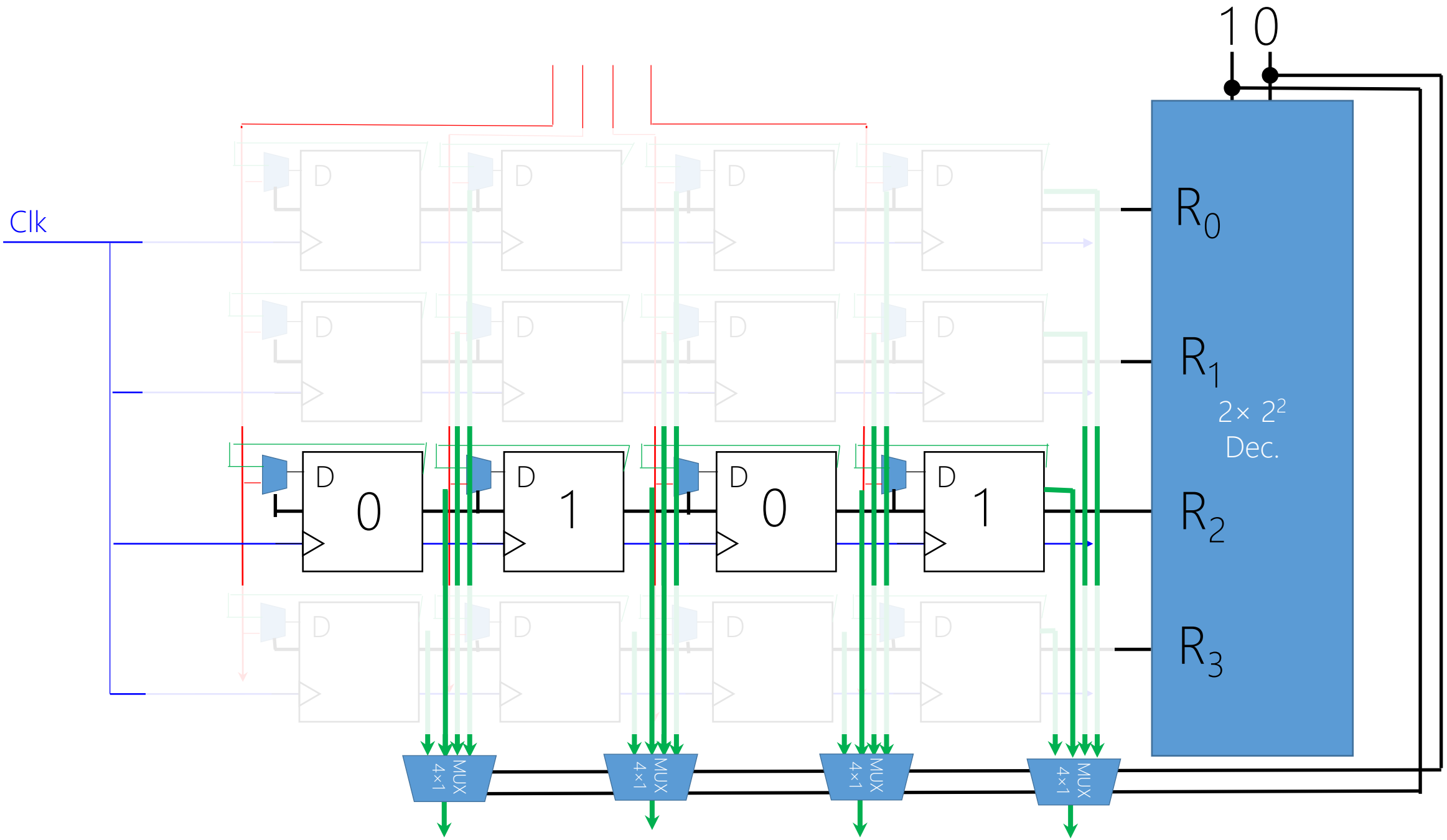


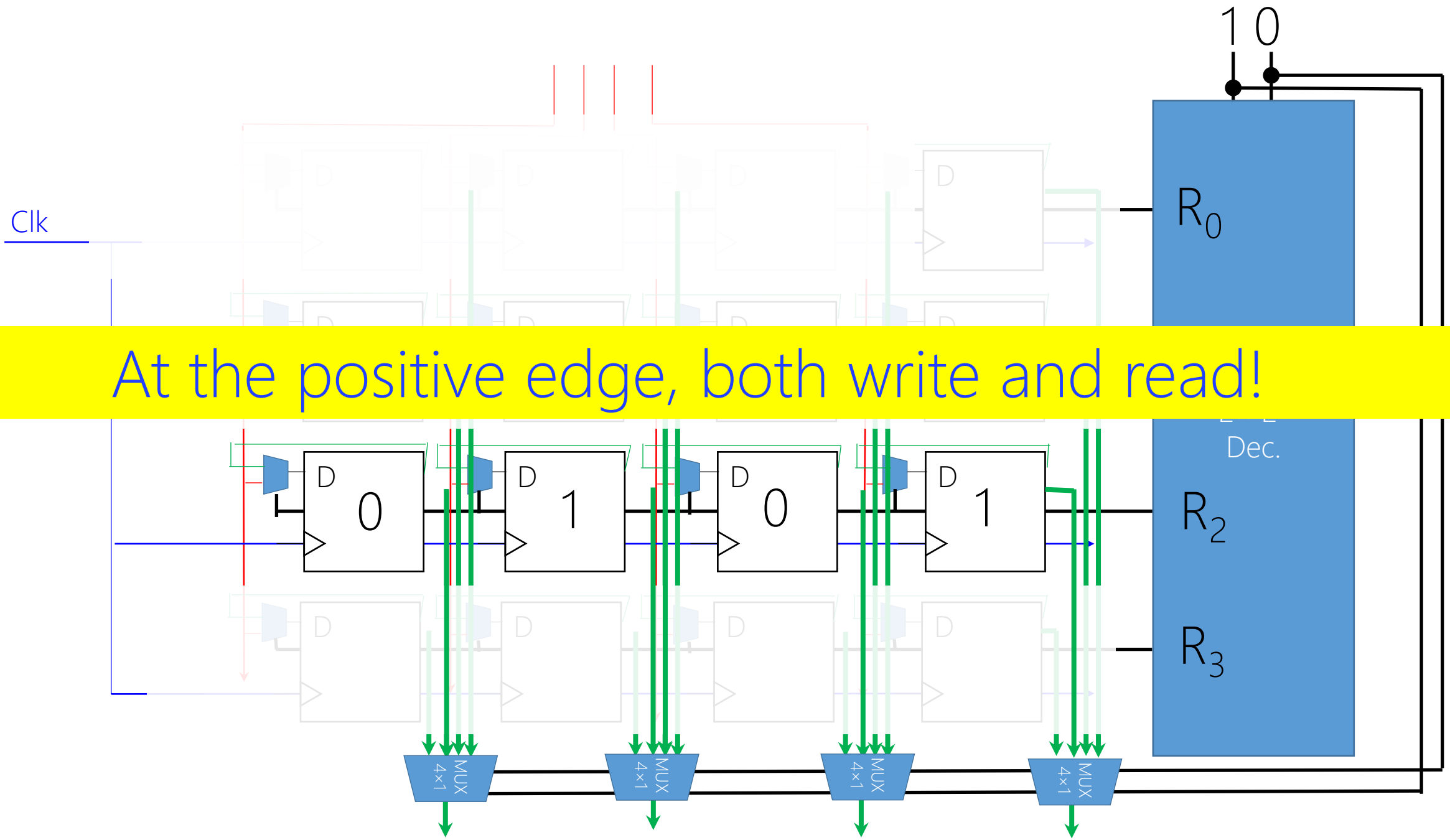

n-bit Data Bus (Write)
$2^m \times$ n-bit Memory Module
m-bit Address Bus
n-bit Data Bus (Read)


1 0
Clk
D
D
D
D
D
D
D
D
D
D
D
D
D
D
D
D
R_0
R_1
2×2^2
Dec.
R_2
R_3


0 1
Clk
D
D
D
D
R_0
R_1
2×2^2
Dec.
R_2
R_3

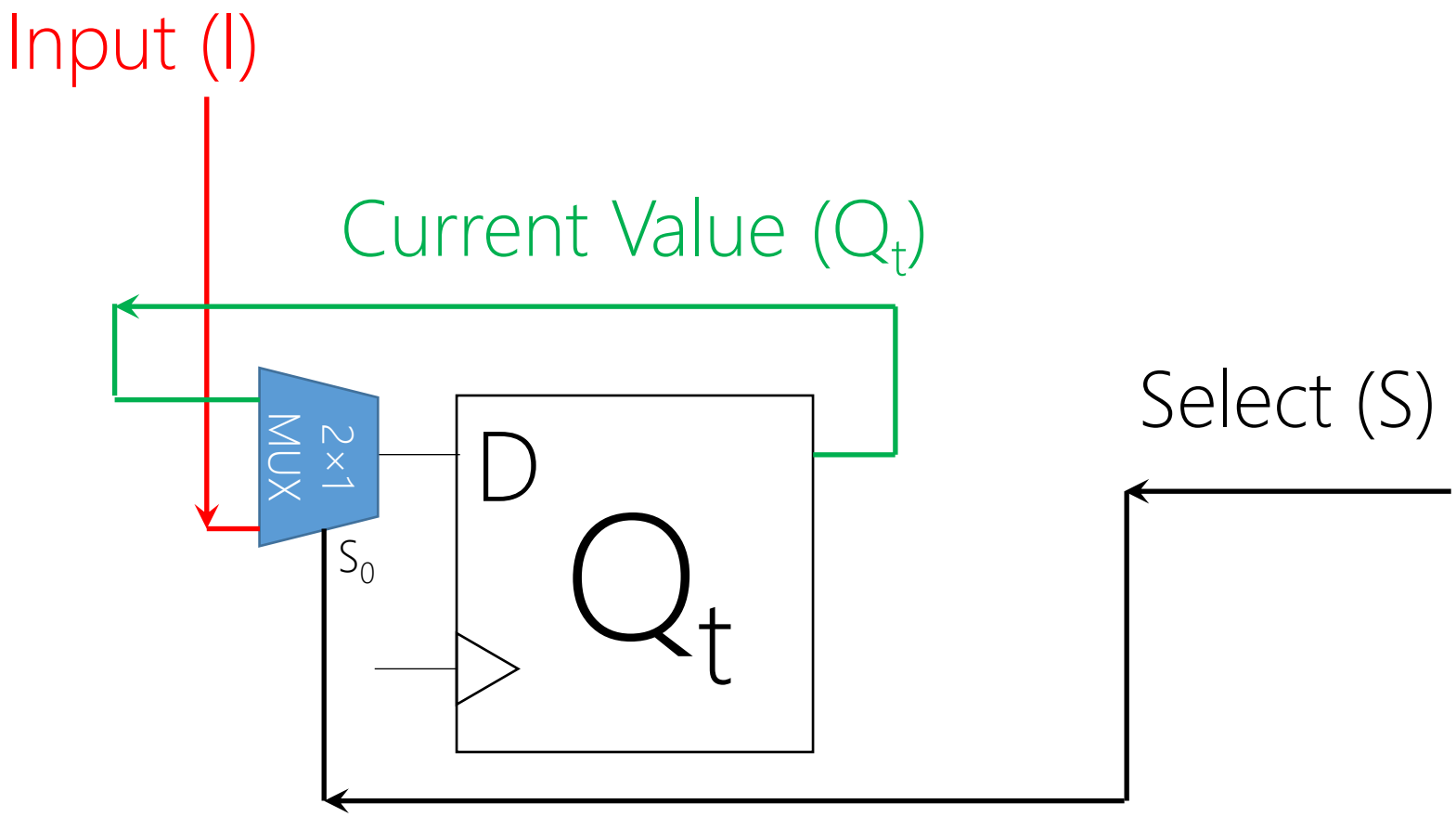

Clk
D
0
1
MUX
4×1
R_0
R_1
R_2
R_3
2× 2^2
Dec.

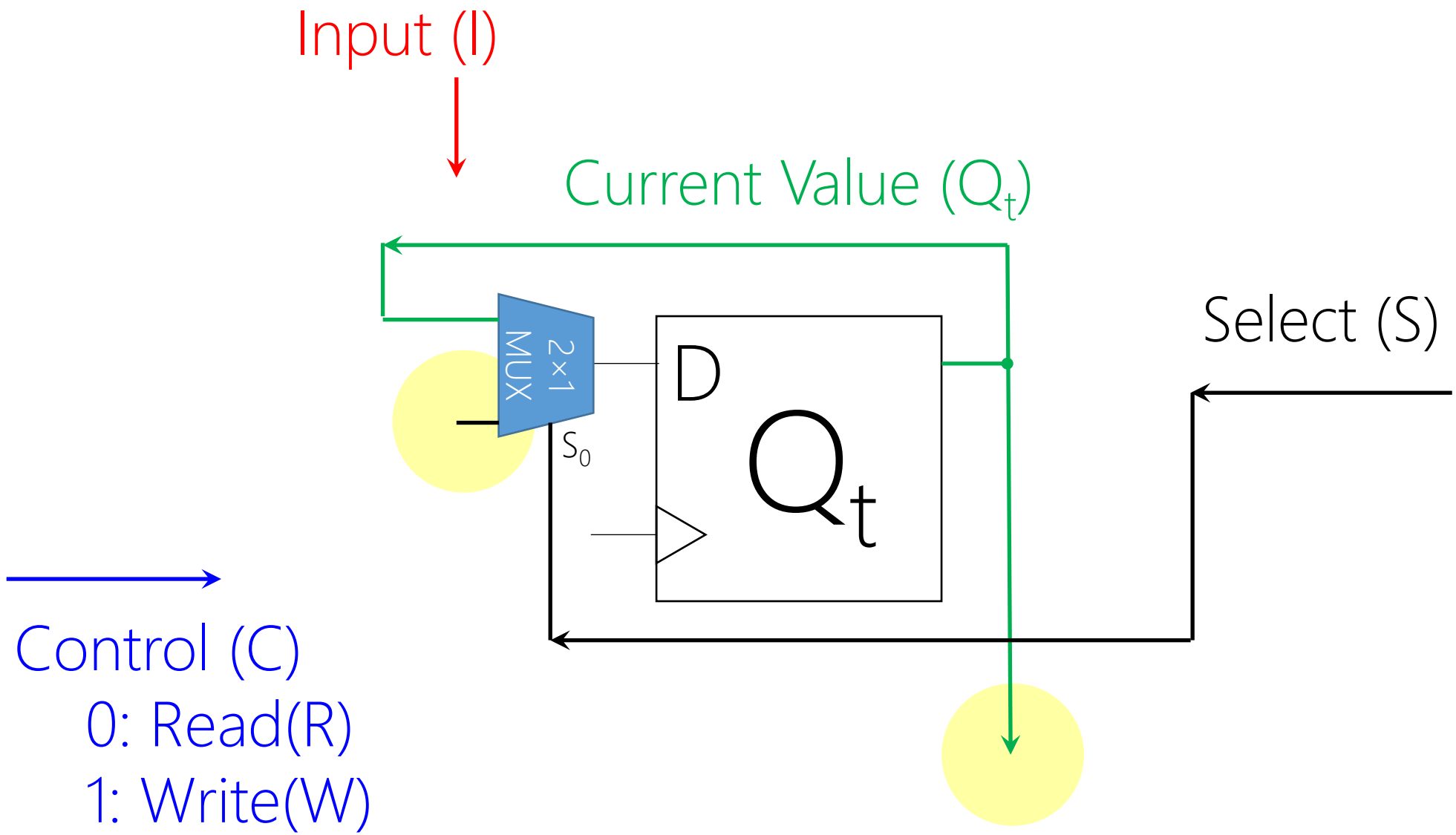

1 0
Clk
D
0
1
R_0
R_1
R_2
R_3
2× 2^2
Dec.
MUX
4×1

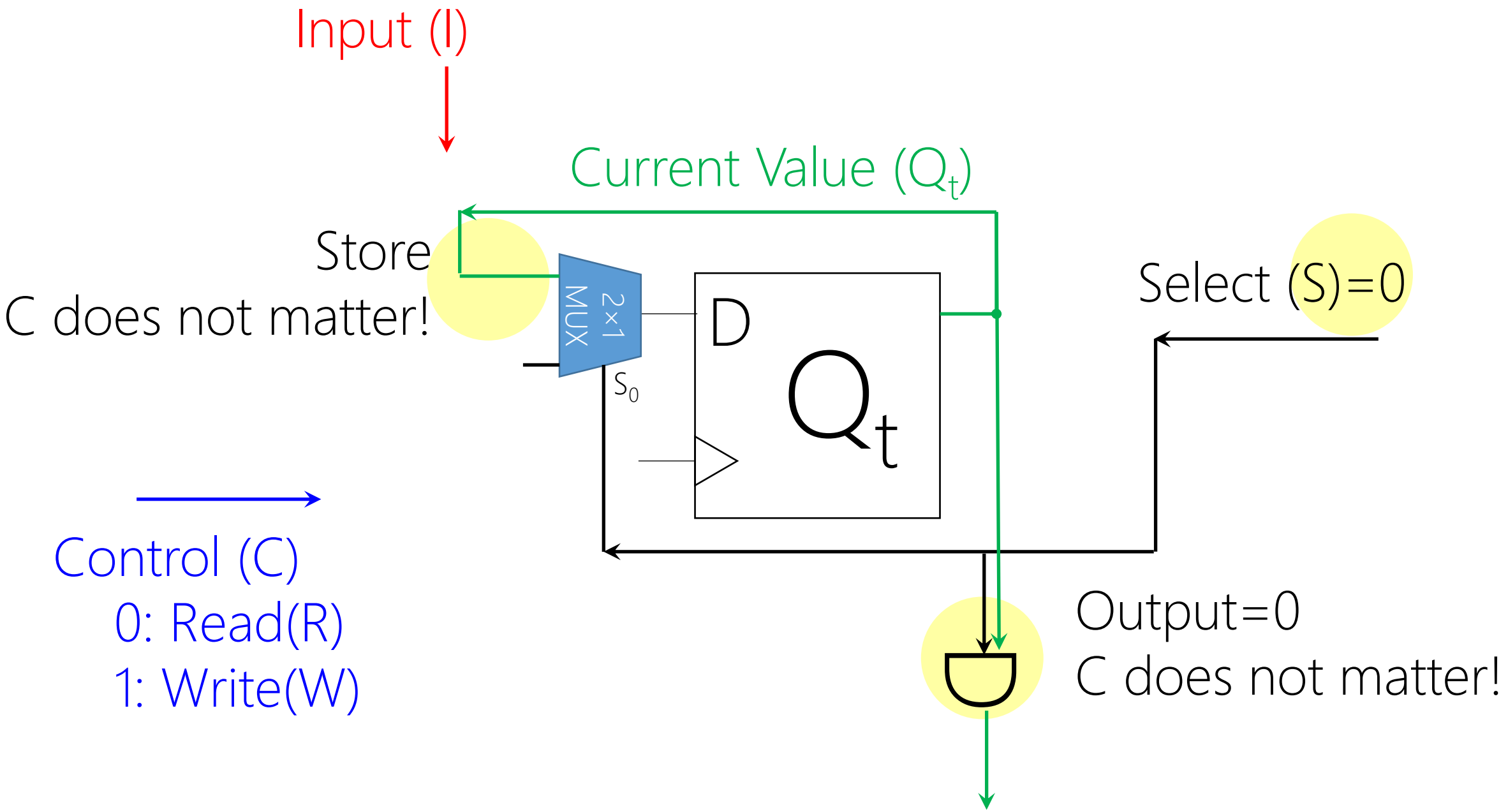


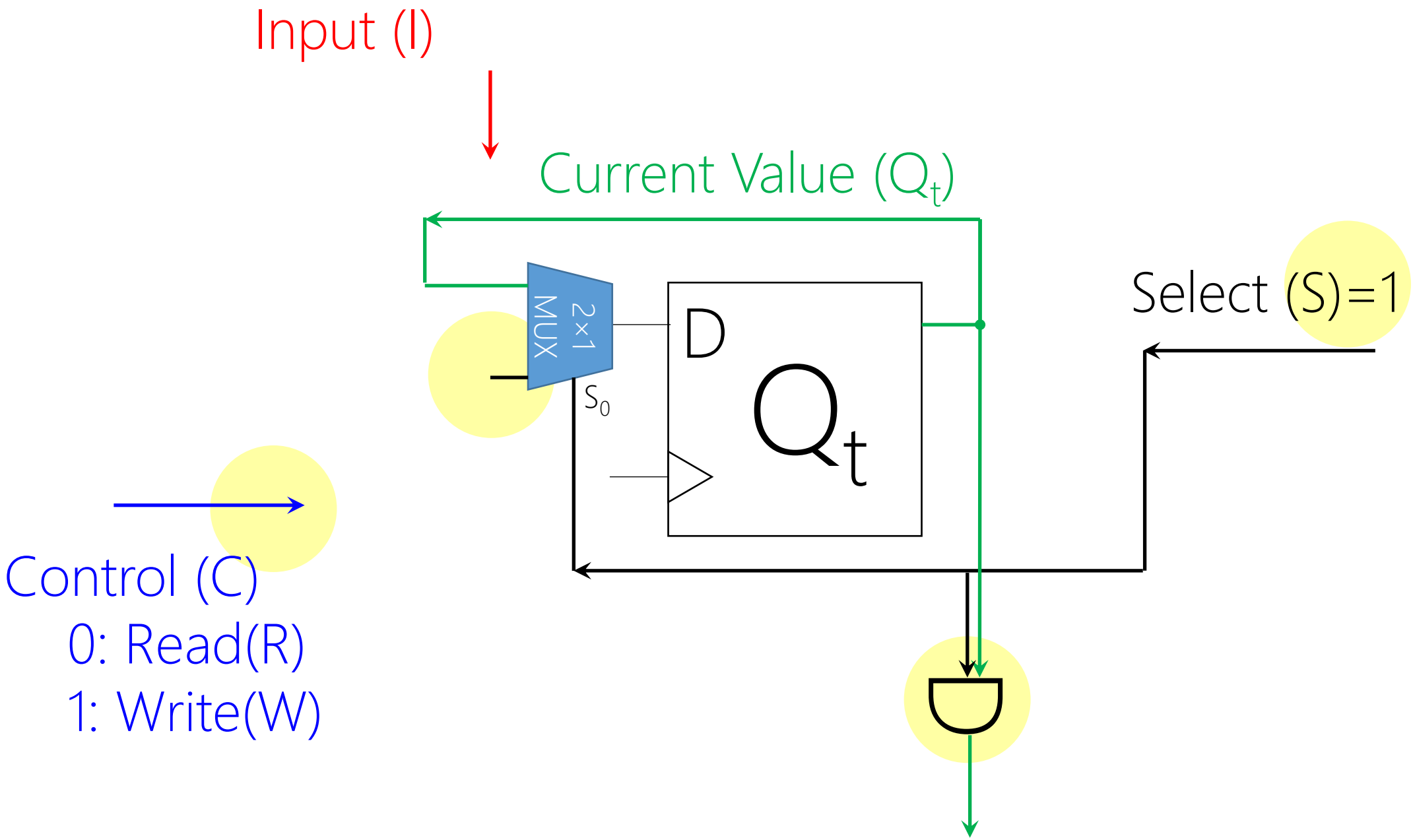
At the positive edge, both write and read!

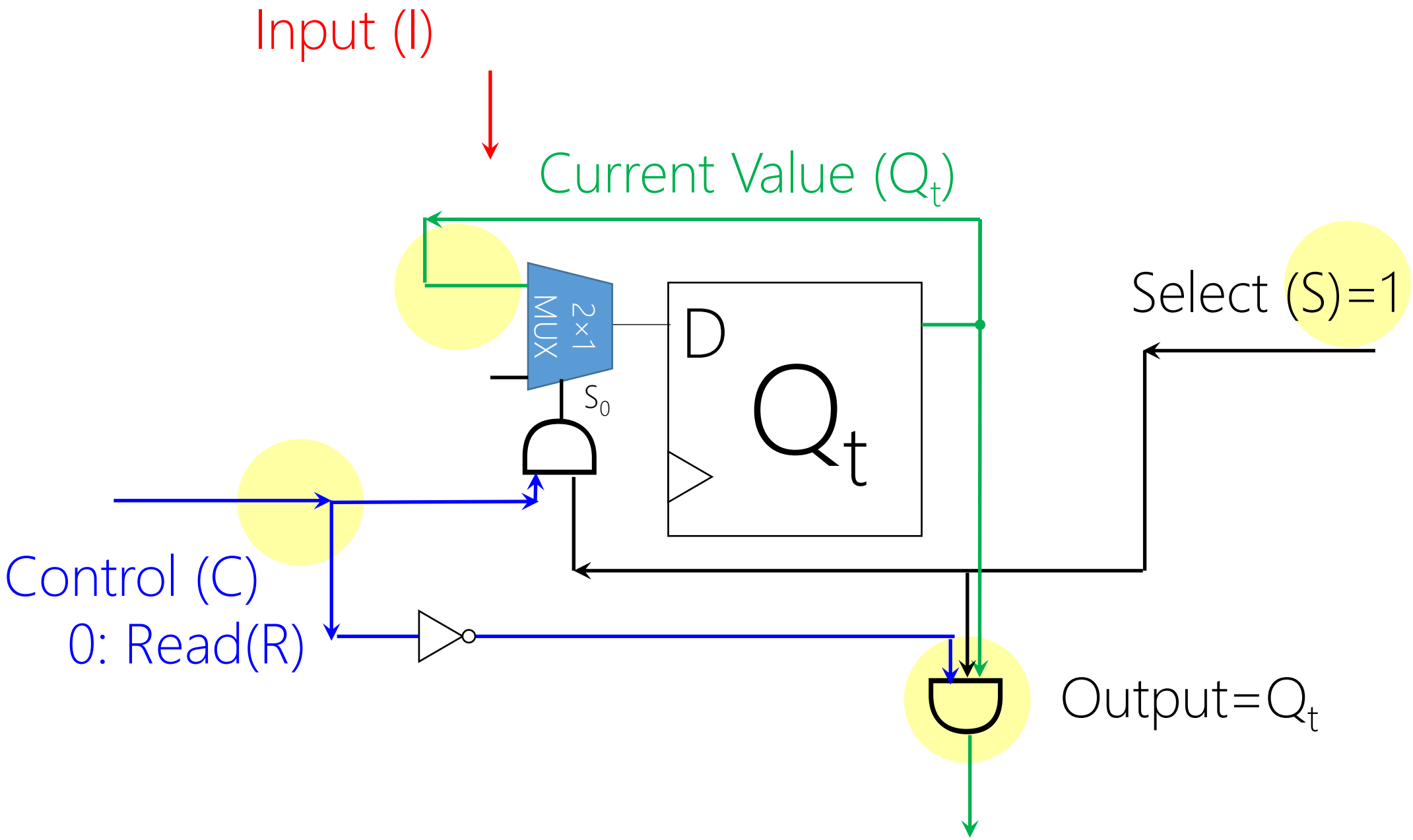
Alternative Design

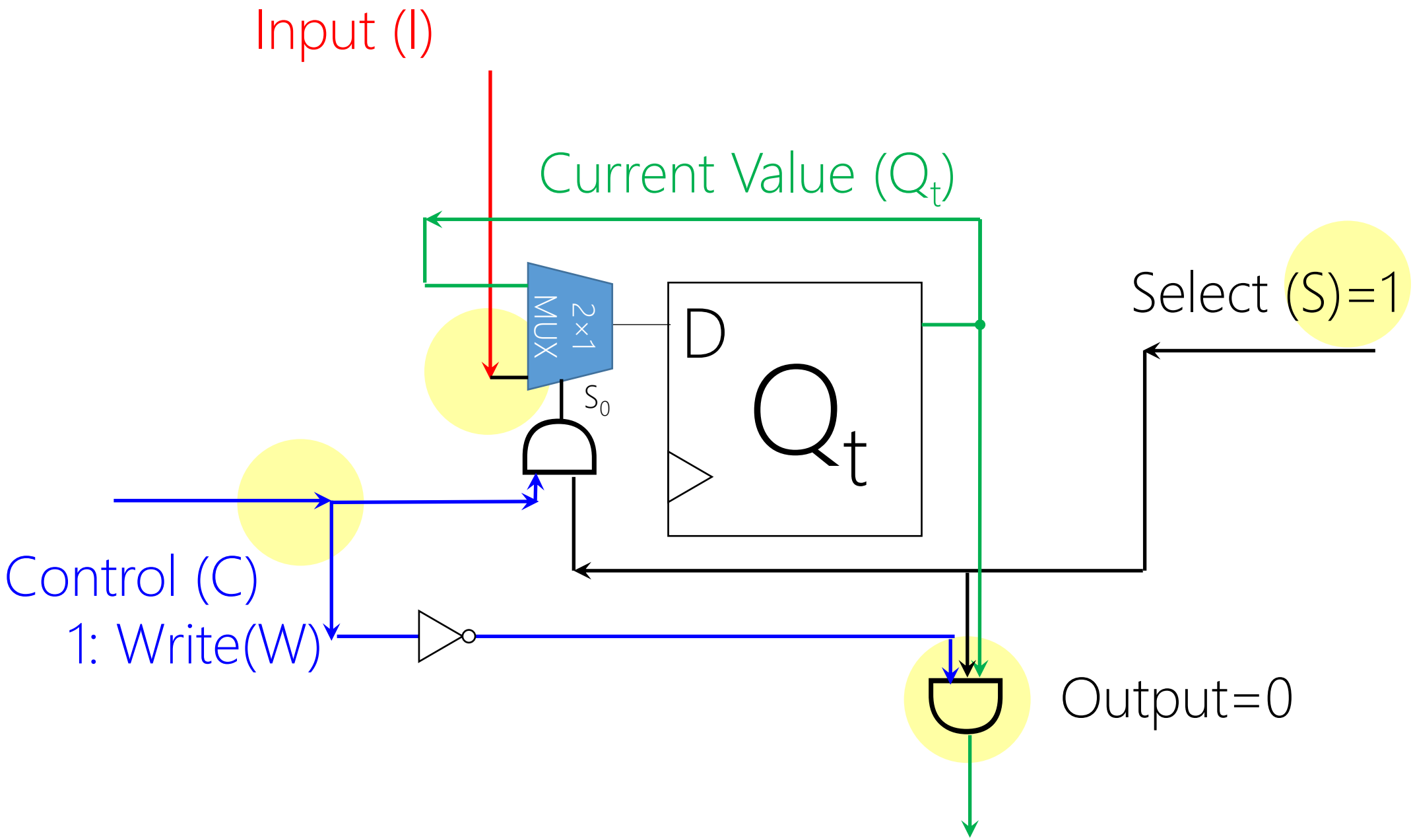

Input (I)
Current Value (Q_t)
Select (S)
2×1 MUX
S_0
D
Q_t

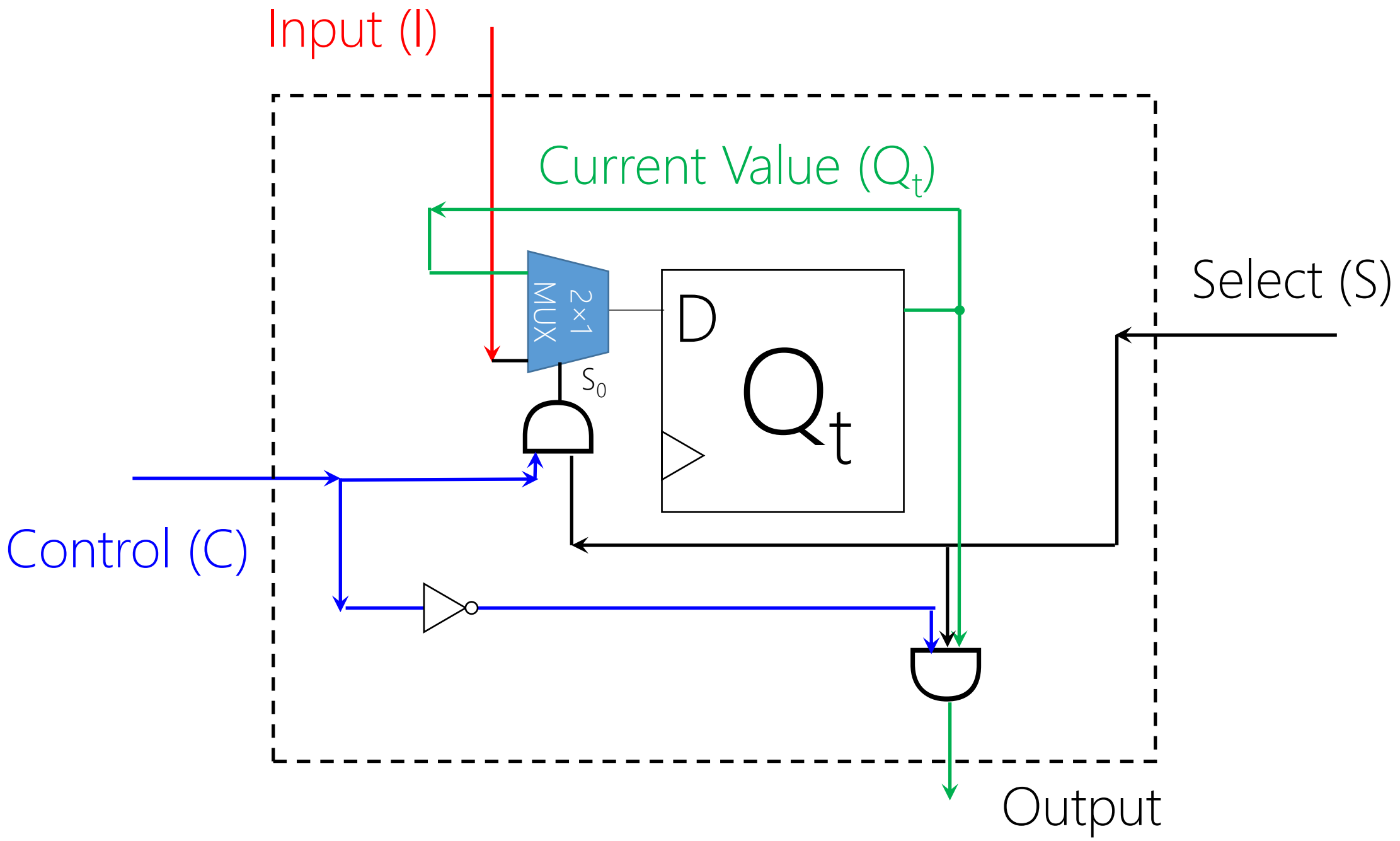

Input (I)
Current Value (Q_t)
2×1 MUX
S_0
D
Q_t
Select (S)
Control (C)
0: Read(R)
1: Write(W)

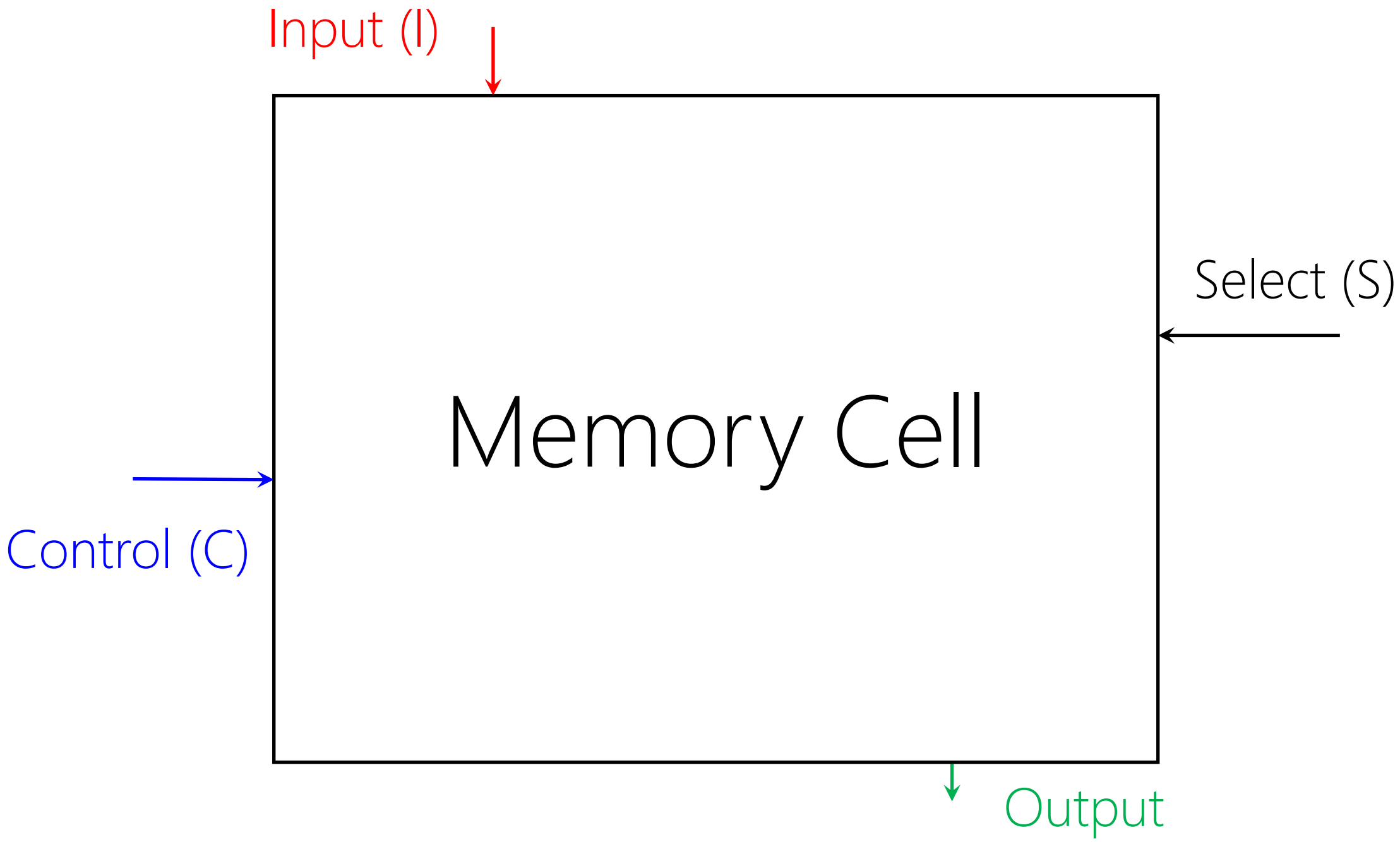

Input (I)
Current Value (Q_t)
Store
C does not matter!
2×1 MUX
S_0
D
Q_t
Select (S)=0
Control (C)
0: Read(R)
1: Write(W)
Output=0
C does not matter!

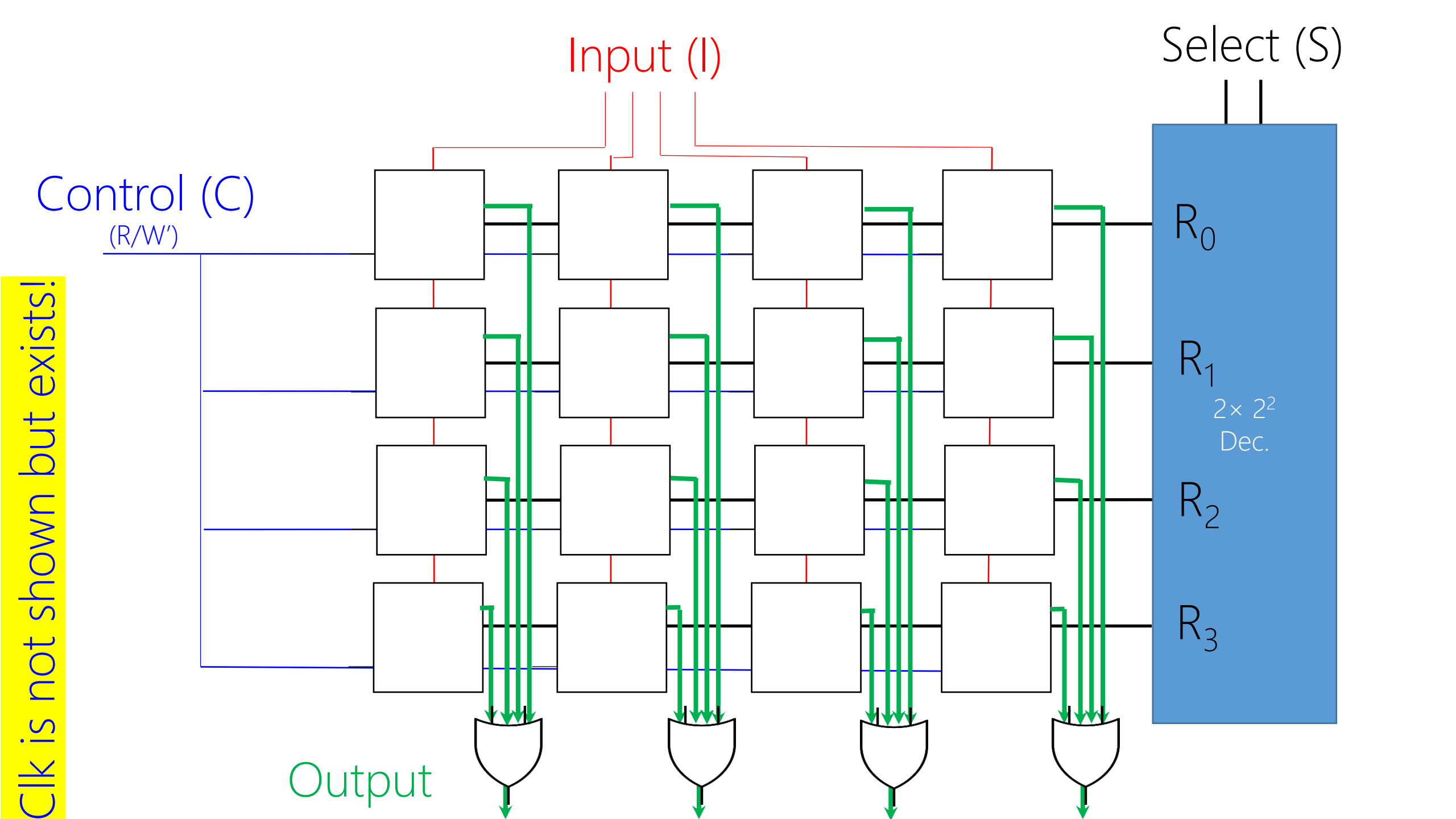

Input (I)
Current Value (Q_t)
2×1 MUX
S_0
D
Q_t
Select (S)=1
Control (C)
0: Read(R)
1: Write(W)

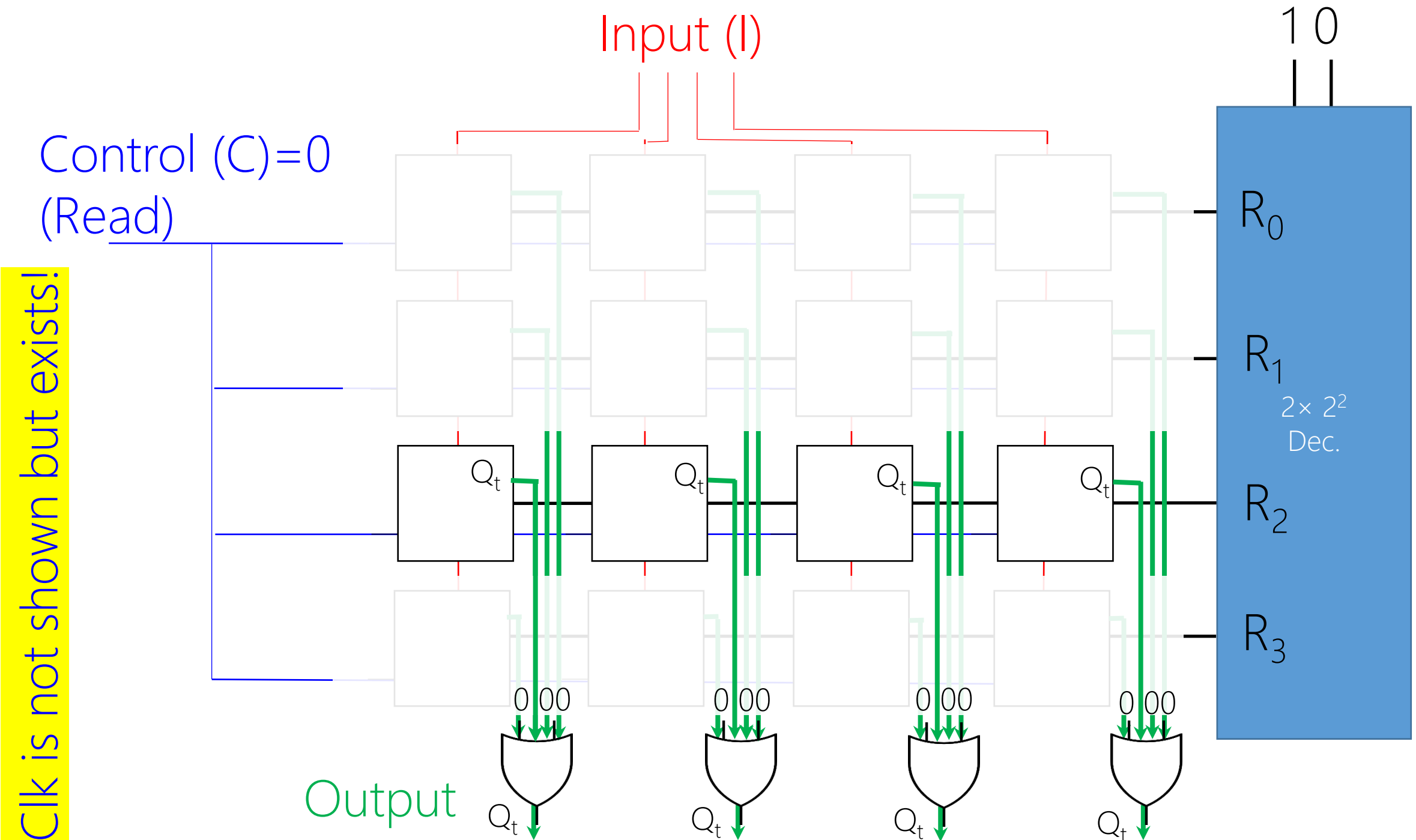

Input (I)
Current Value (Q_t)
2×1 MUX
S_0
D
Q_t
Select (S)=1
Control (C)
0: Read(R)
Output=Q_t

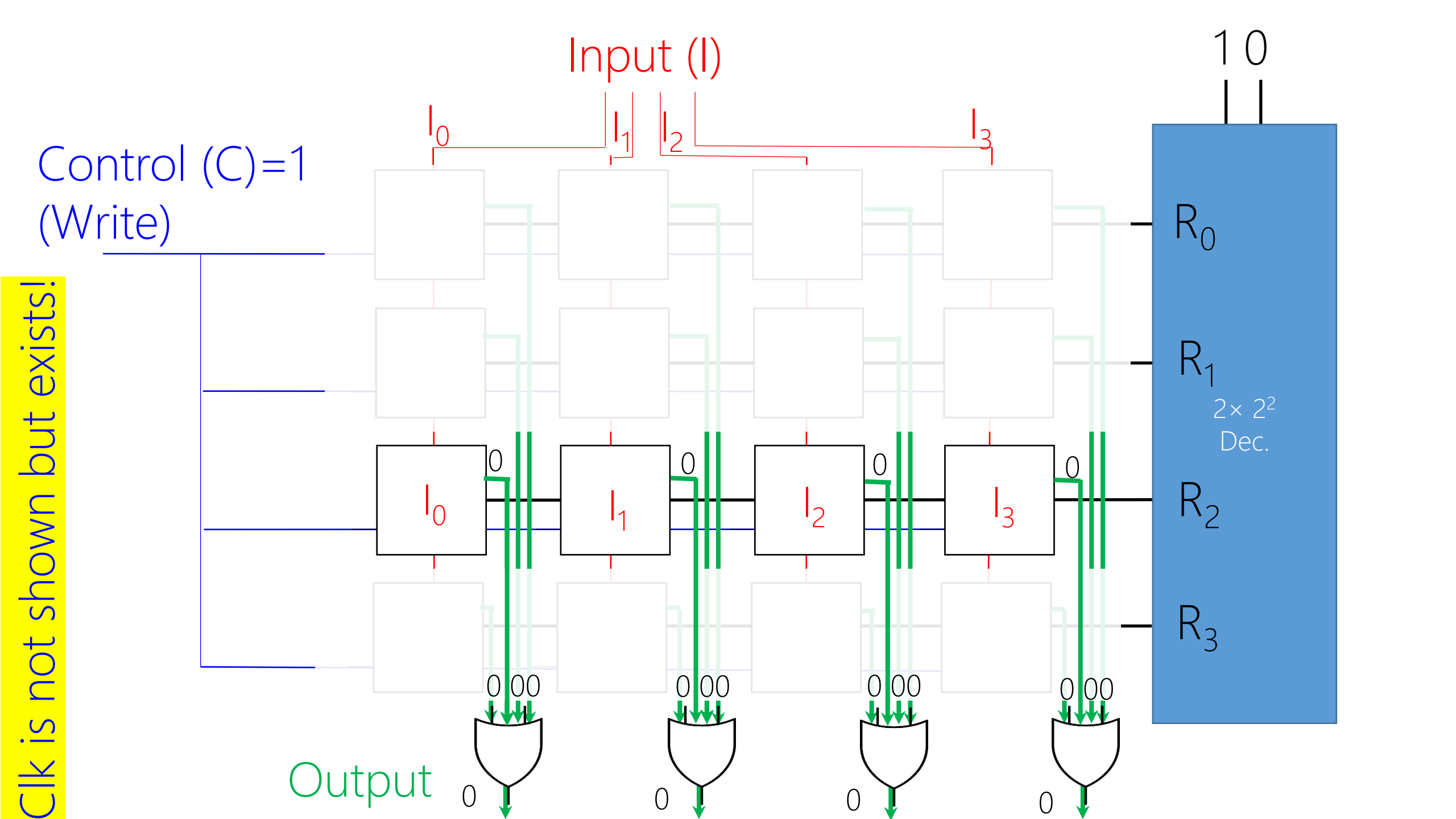

Input (I)
Current Value (Q_t)
2×1 MUX
S_0
D
Q_t
Select (S)=1
Control (C)
1: Write(W)
Output=0

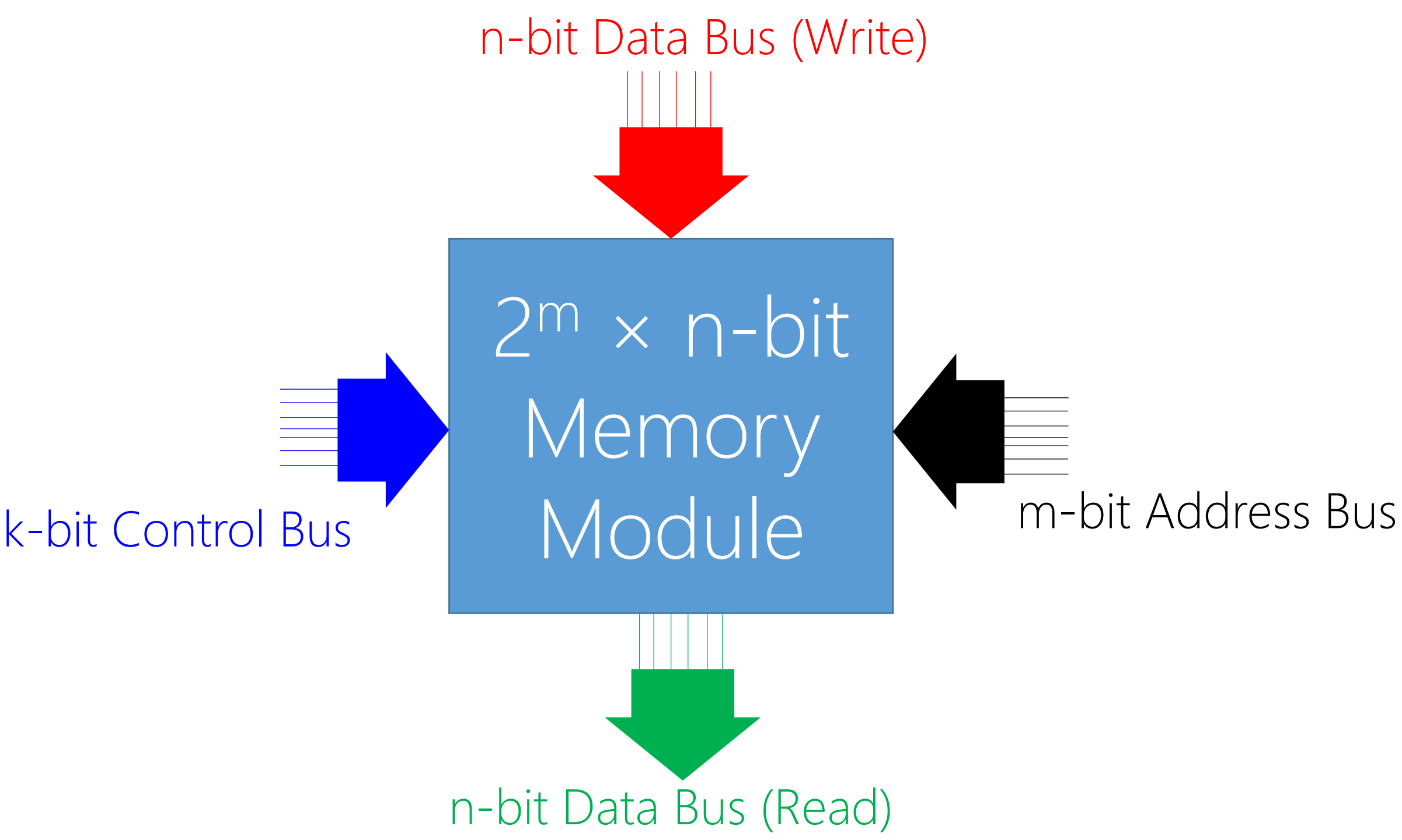

Input (I)
Current Value (Q_t)
Select (S)
2×1 MUX
S_0
D
Q_t
Control (C)
Output


Input (I)
Memory Cell
Select (S)
Control (C)
Output


Input (I)
Select (S)
Control (C)
(R/W')
Clk is not shown but exists!
R_0
R_1
R_2
R_3
2×2^2
Dec.
Output

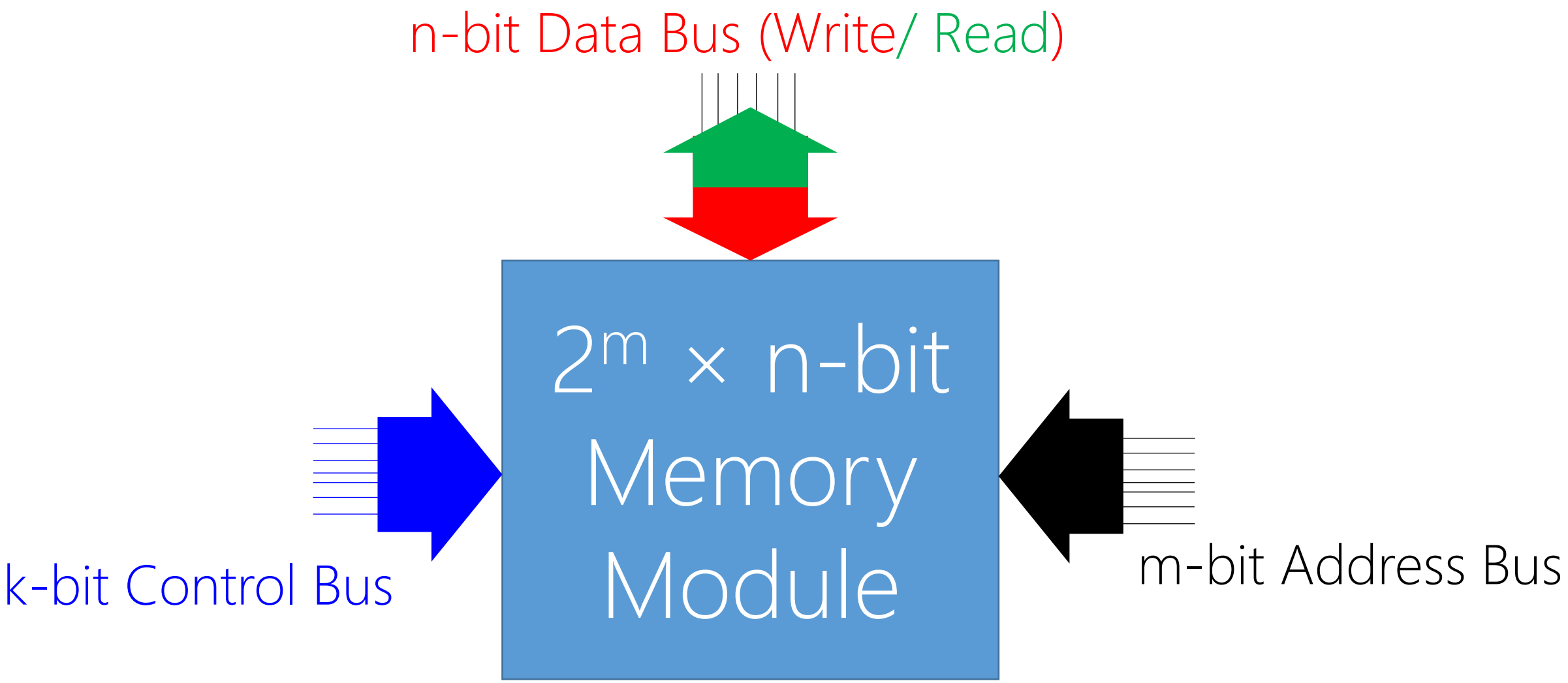

Input (I)
1 0
Control (C)=0
(Read)
Clk is not shown but exists!
R_0
R_1
2×2^2
Dec.
R_2
R_3
Q_t
Q_t
Q_t
Q_t
0 00
0 00
0 00
0 00
Output
Q_t
Q_t
Q_t
Q_t


Input (I)
1 0
I_0
I_1
I_2
I_3
Control (C)=1
(Write)
Clk is not shown but exists!
R_0
R_1
R_2
R_3
2× 2^2
Dec.
0
0 00
Output


n-bit Data Bus (Write)
$2^m \times$ n-bit Memory Module
k-bit Control Bus
m-bit Address Bus
n-bit Data Bus (Read)

Input and Output

Same Data Bus vs. Another Data Bus


n-bit Data Bus (Write/ Read)
$2^m \times$ n-bit Memory Module
k-bit Control Bus
m-bit Address Bus

Access
Random vs. Sequential

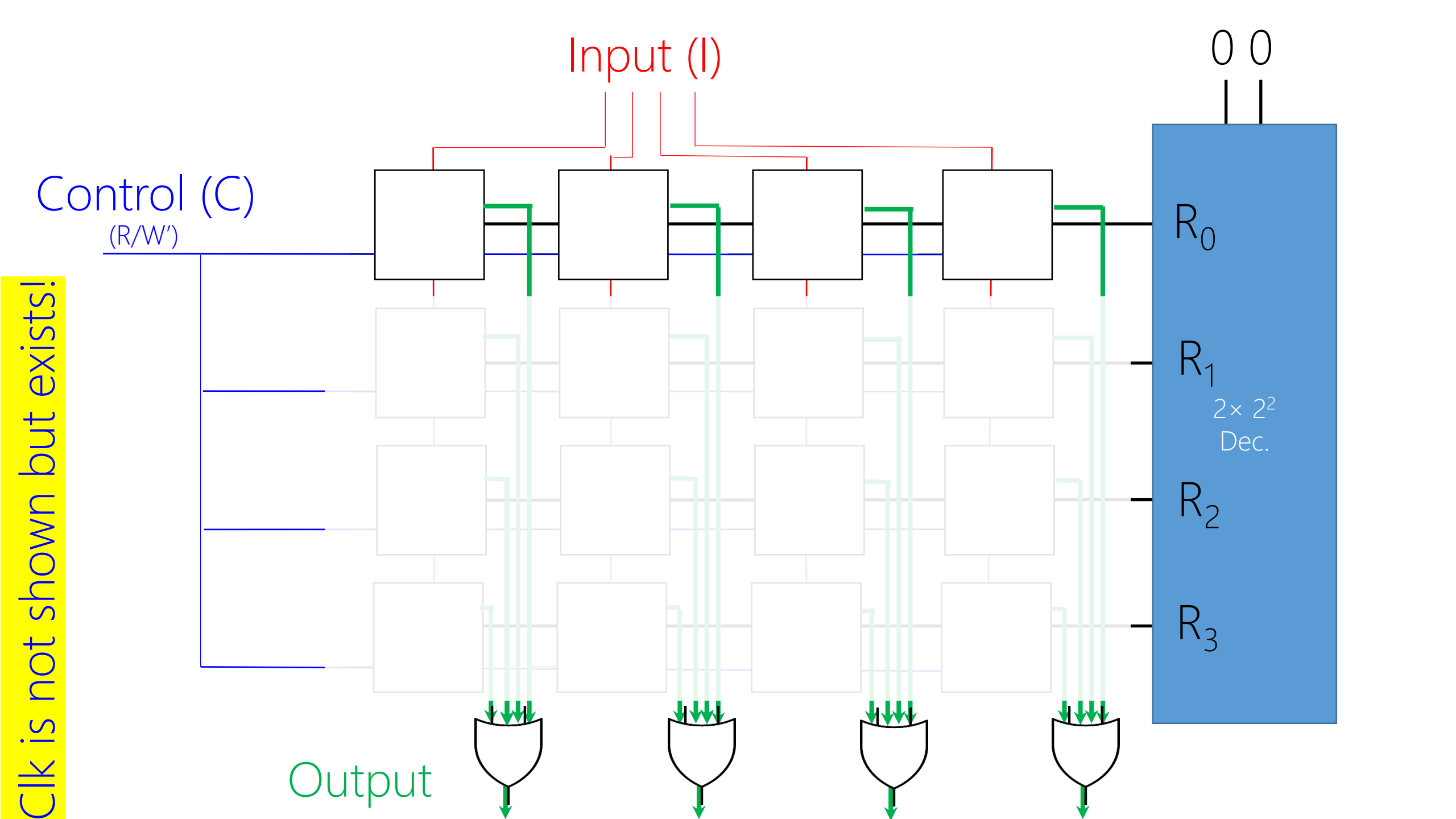
Access
Random vs. Sequential

Random: regardless of location, always fixed Δt.
Sequential: closer location sooner, farther location longer!

Access
Random vs. Sequential

How long does it take to select a memory location (register)?
E.g., if address bus loads 00,

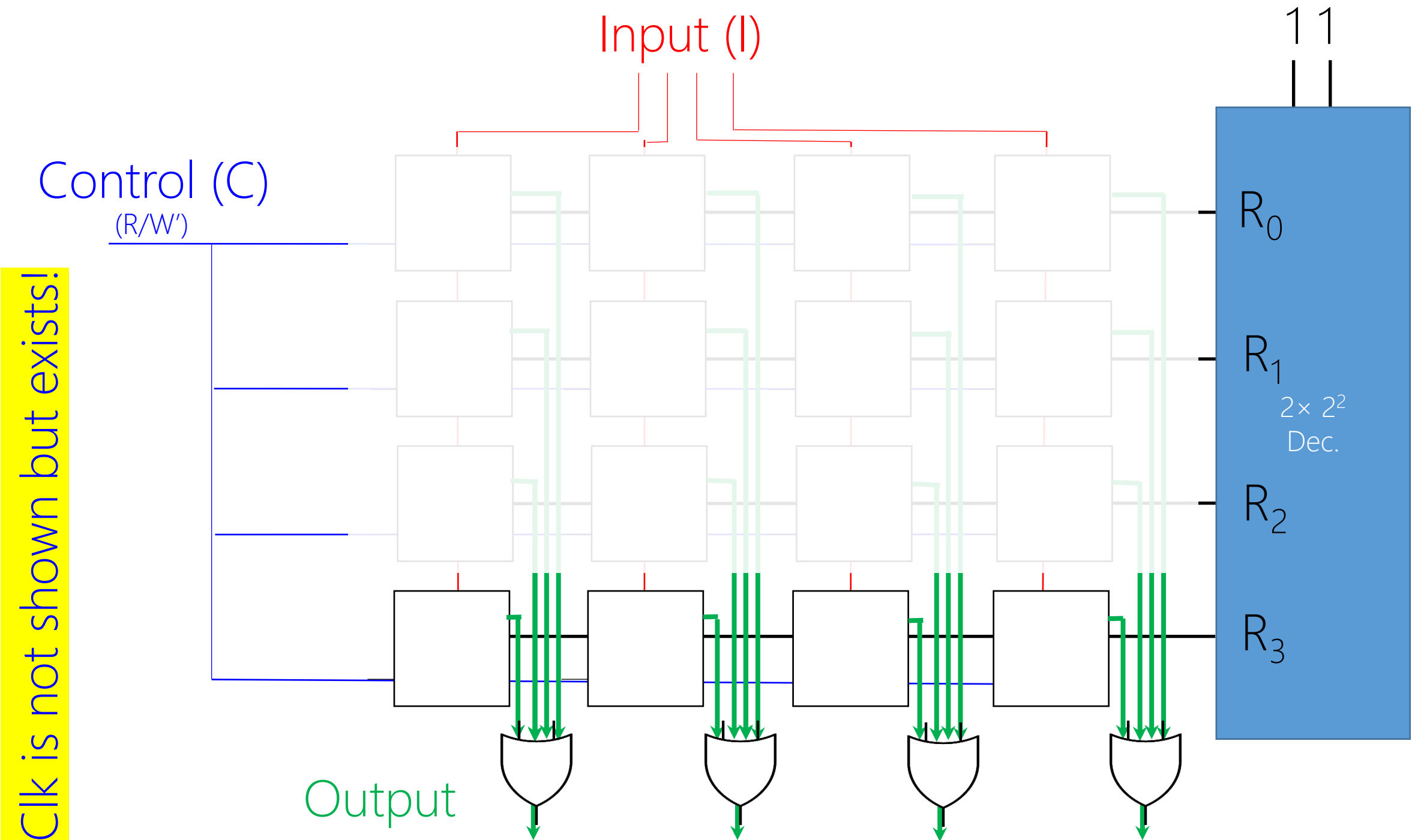
- how long is the wait to read this memory location?
- how long is the wait to write this memory location?


Input (I)
0 0
Control (C)
(R/W')
R_0
R_1
R_2
R_3
2×2^2
Dec.
Output
Clk is not shown but exists!

Access
Random vs. Sequential

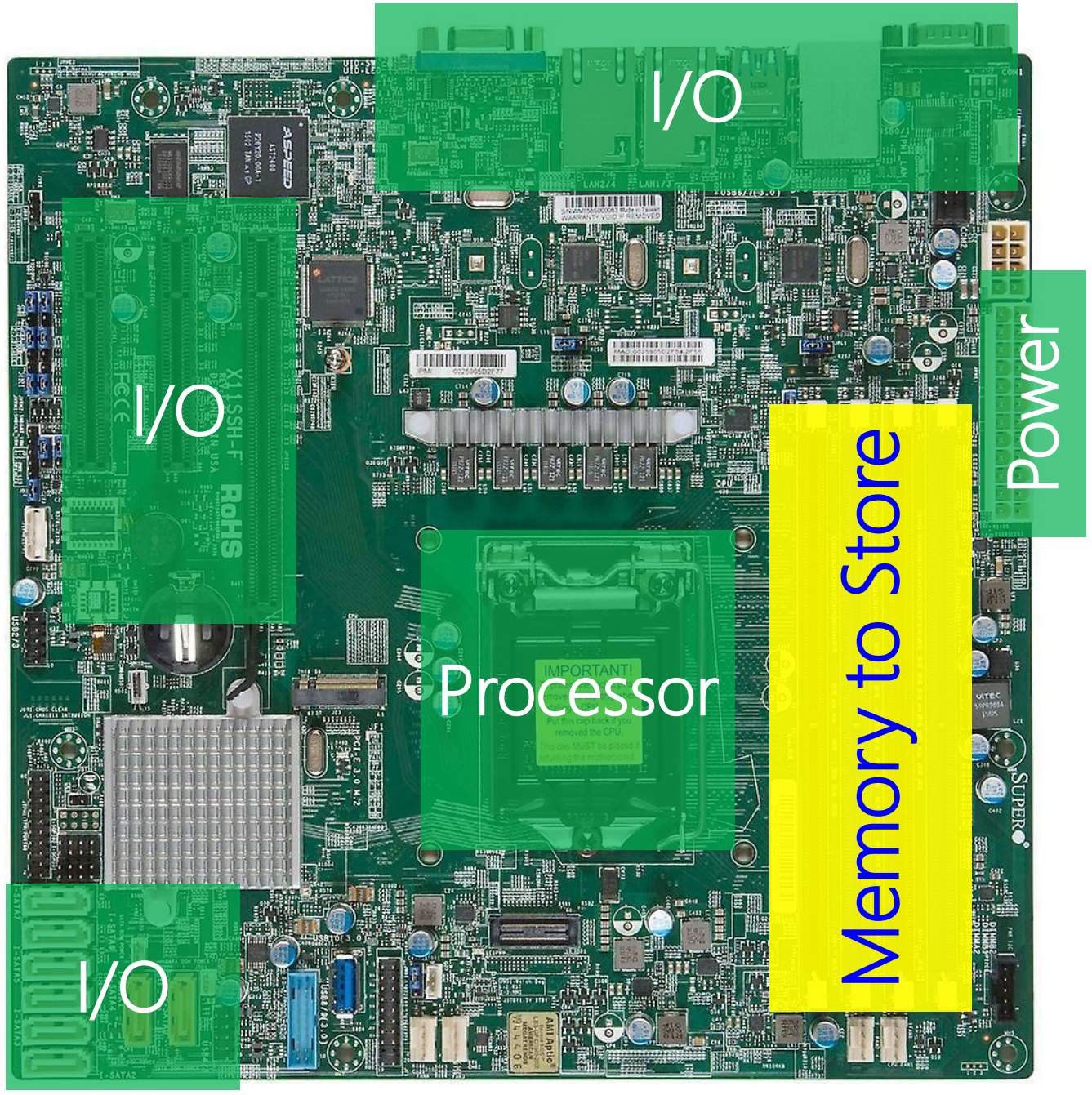
How long does it take to select a memory location (register)?
E.g., if address bus loads 11,

- how long is the wait to read this memory location?
- how long is the wait to write this memory location?


Input (I)
11
Control (C)
(R/W')
Clk is not shown but exists!
R_0
R_1
2×2^2
Dec.
R_2
R_3
Output

Random Access Memory (RAM)

Regardless of memory location, always fixed Δt.

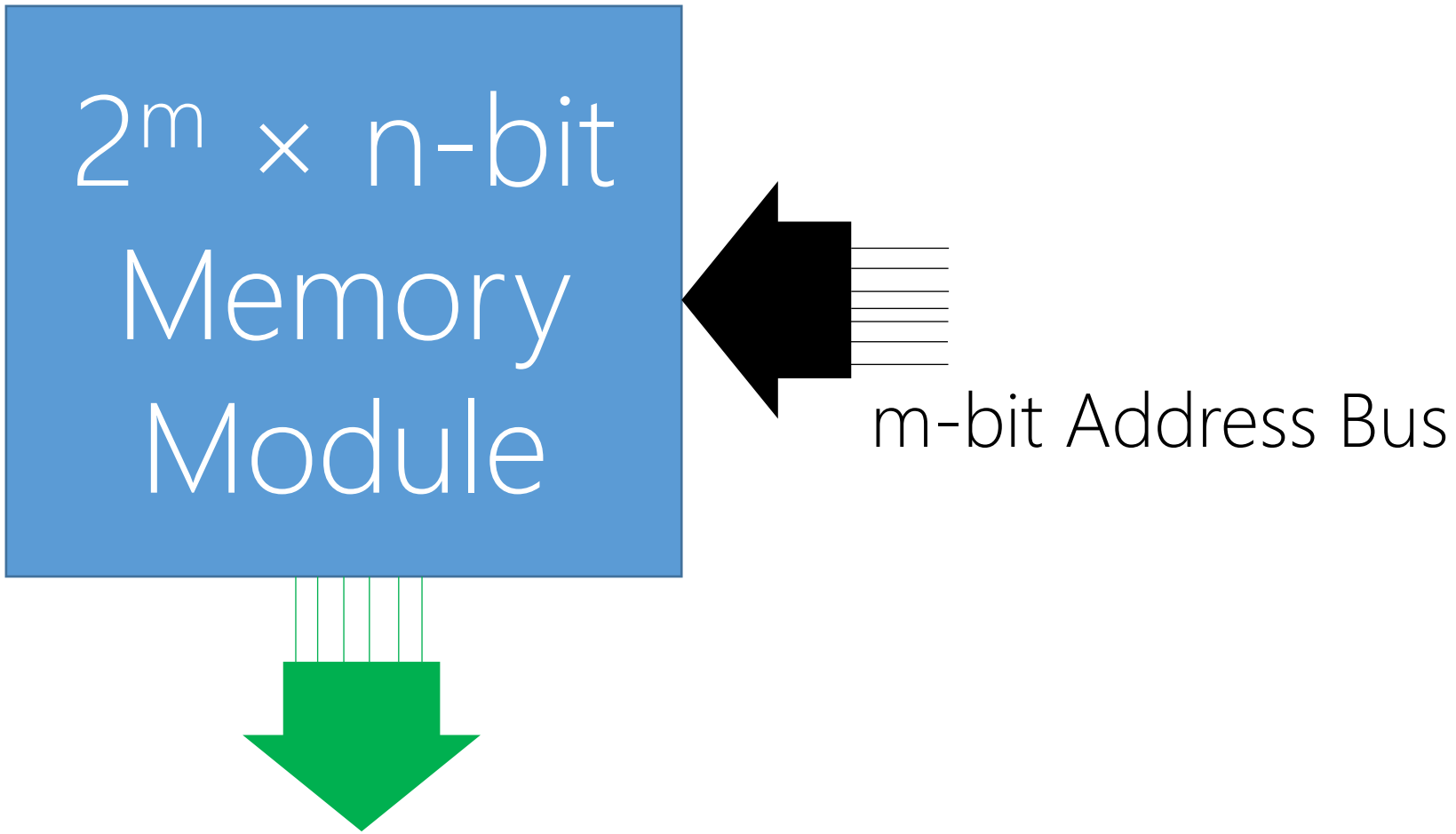

I/O
I/O
Power
Memory to Store
Processor
I/O

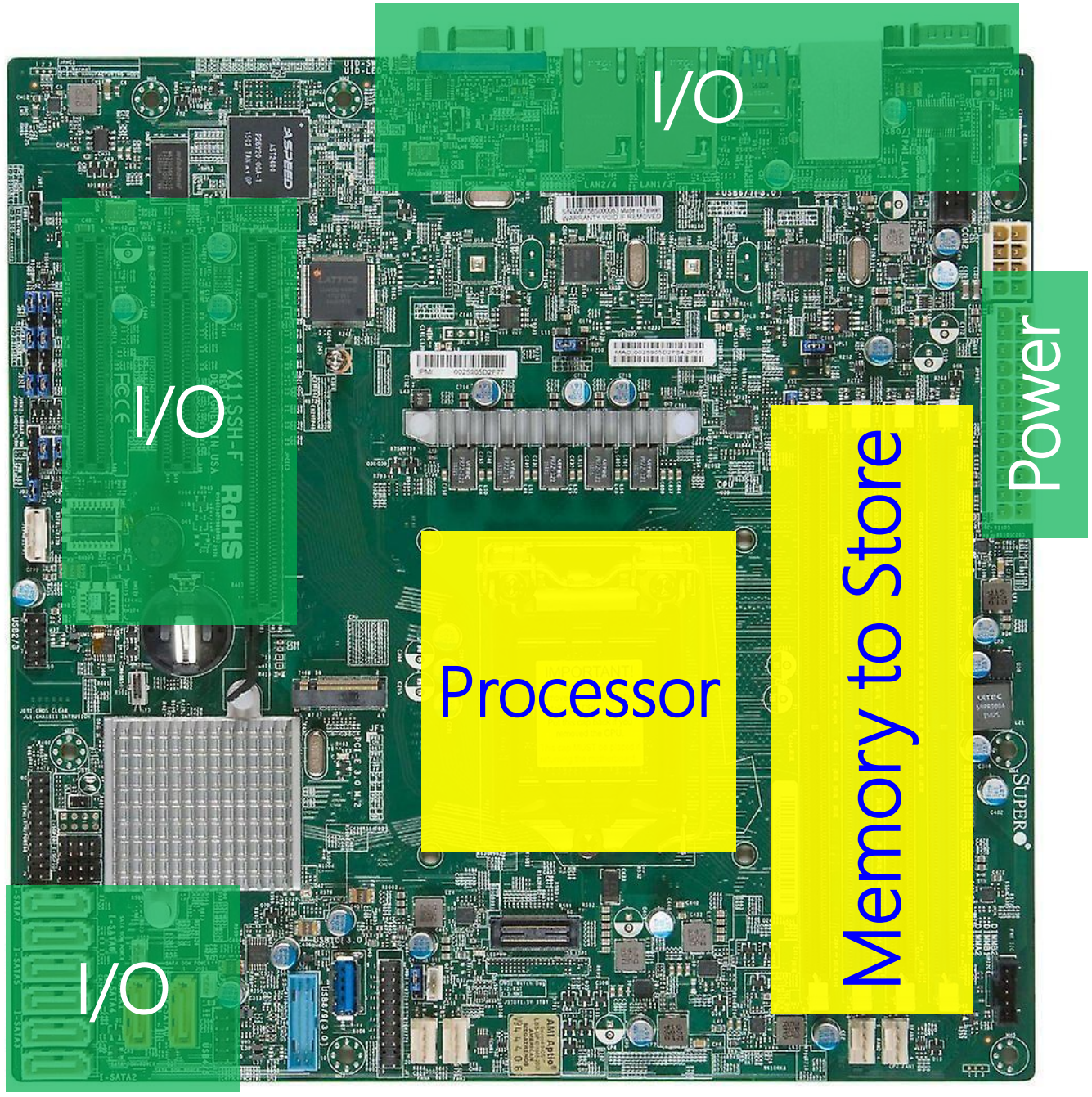
Random Access Read-only Memory (ROM)

They don’t have control bus for writing!

To store static data.

Still can write (update) but via physical procedures.


$2^m \times$ n-bit Memory Module
m-bit Address Bus
n-bit Data Bus (Read)


I/O
I/O
Power
Processor
Memory to Store
I/O