COMPUTER SYSTEMS AND ORGANIZATION

Part 1

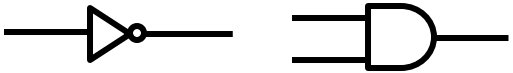
Daniel G. Graham PhD

September 11, 2023

ENGINEERING

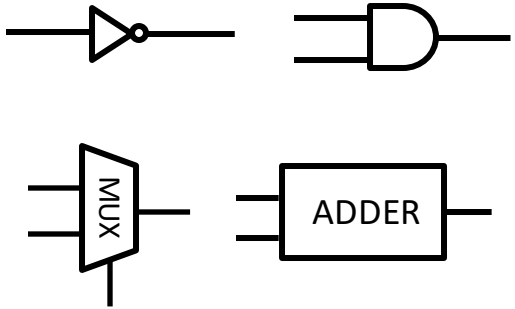
REVIEW

- ✓ What are logic gates?



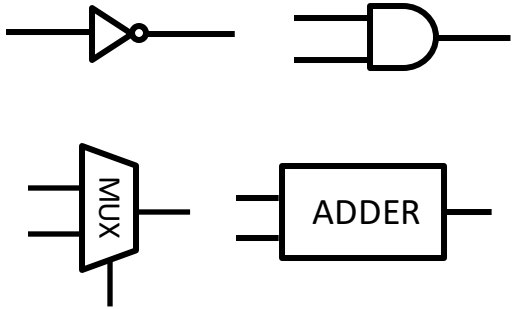
REVIEW

- ✓ What are logic gates?
- ✓ How to make circuits like Multiplexers, Adders?



REVIEW

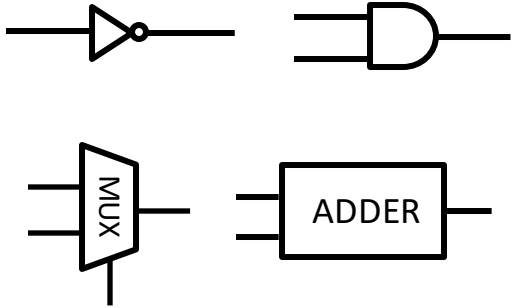
- ✓ What are logic gates?
- ✓ How to make circuits like Multiplexers, Adders?
- ✓ How to represent numbers in different formats?



Binary: 0110 Hex:0xAF23

REVIEW

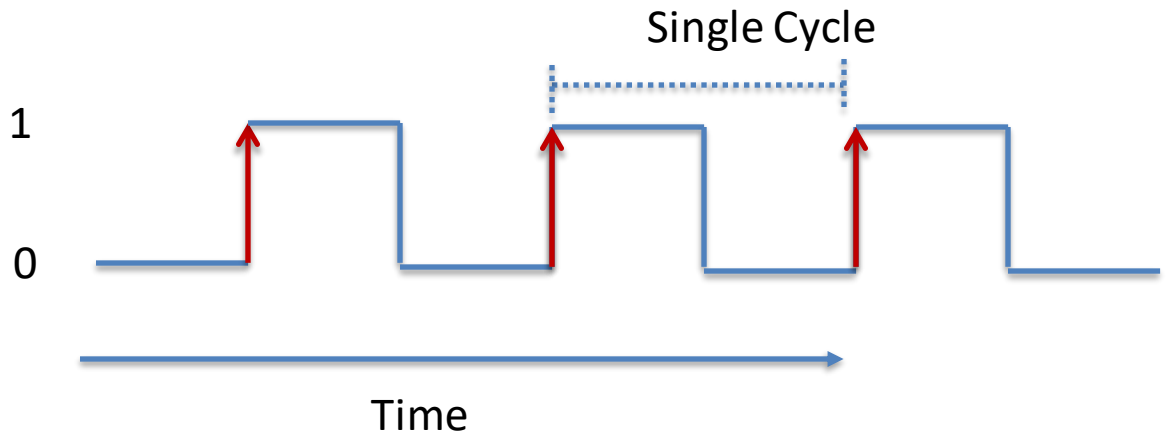
- ✓ What are logic gates?
- ✓ How to make circuits like Multiplexers, Adders?
- ✓ How to represent numbers in different formats?
- ✓ How to store data in registers?



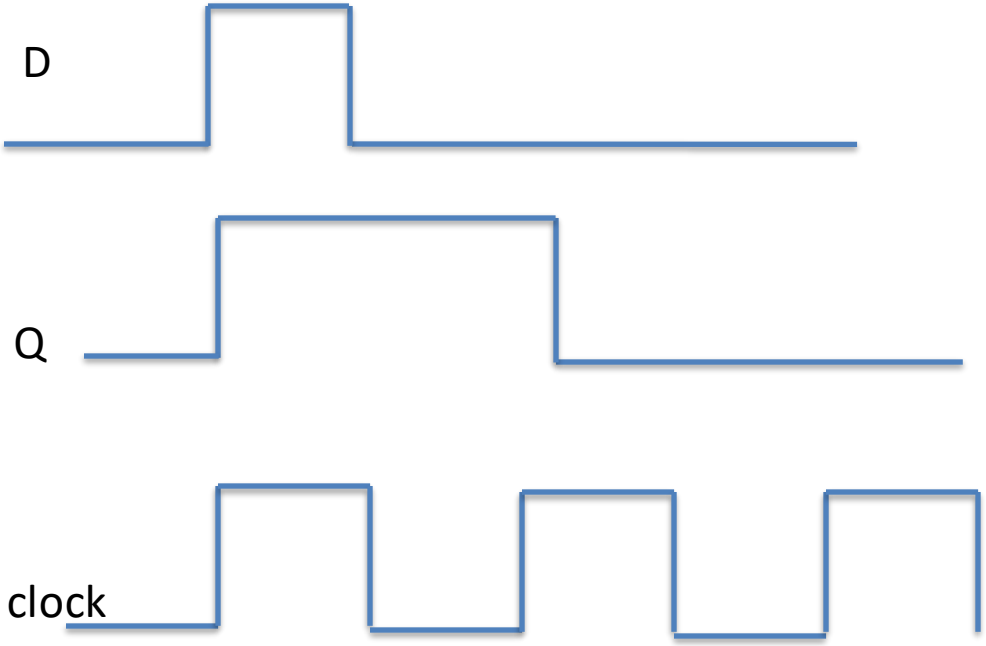
Binary: 0110 Hex:0x6

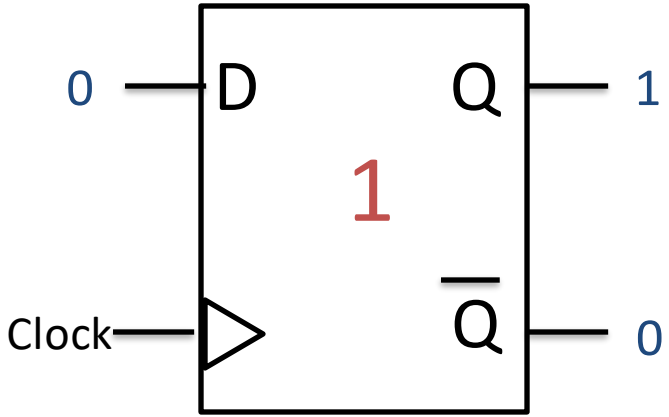
CLOCKS EDGES

Rising Edge (Also called positive edge)

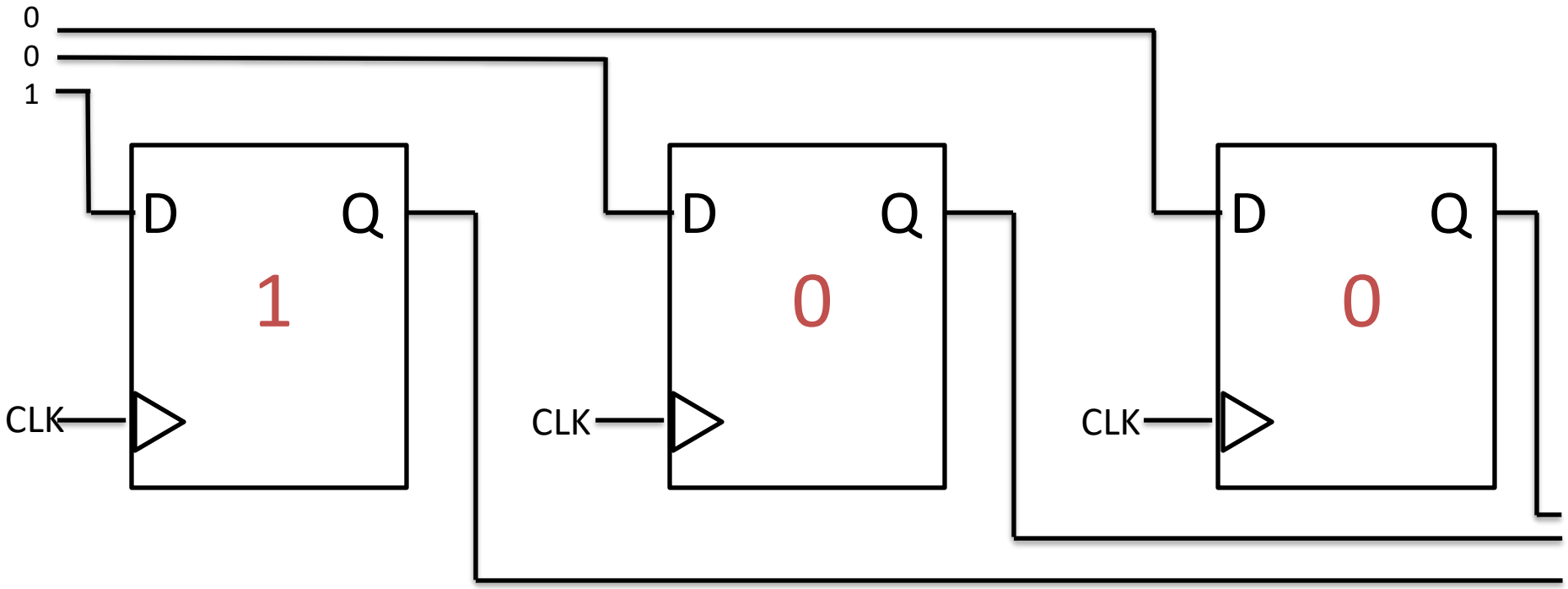


THE FLIP FLOP HOLDS THE VALUE FOR A CLOCK CYCLE





BUILDING A REGISTER FROM FLIP FLOPS



Removed Q (bar) for readability

3-BIT COUNTER

Let's put it all together and build a 3-bit counter

Circuit that counts from

000,

001,

010,

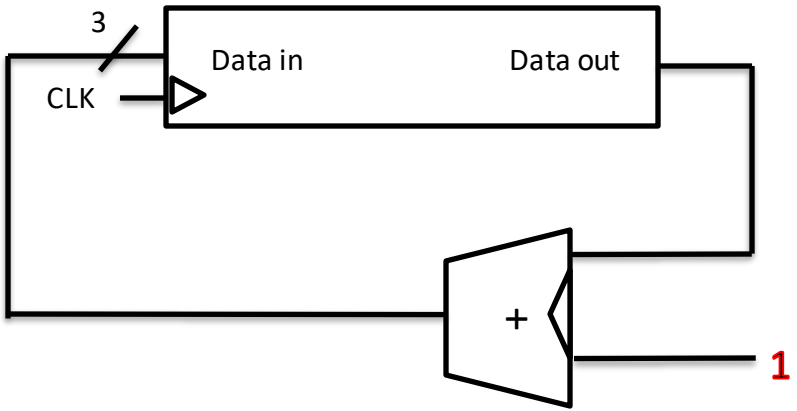
011,

100,

101,

110,

111



TODAY'S LECTURE

1. How do we use registers as building blocks to design a computer?

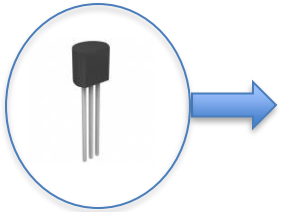
TODAY'S LECTURE

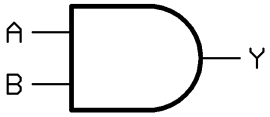
1. How do we use registers as building blocks to design a computer?
2. What is a register file and how to implement it?

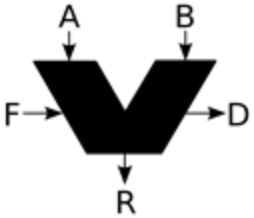
TODAY'S LECTURE

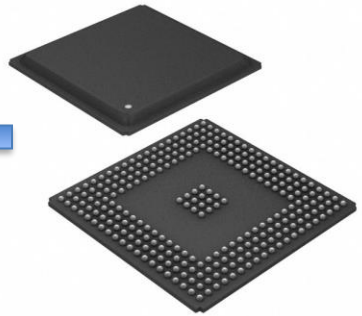
1. How do we use registers as building blocks to design a computer?
2. What is a register file and how to implement it?
3. Other memory components

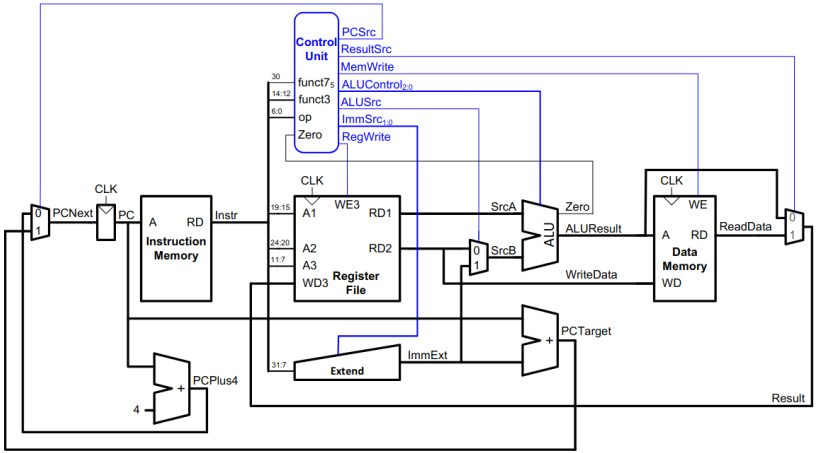
THE MAP (THE MACHINE)





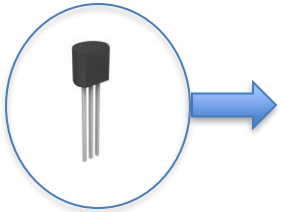


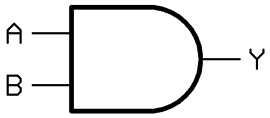


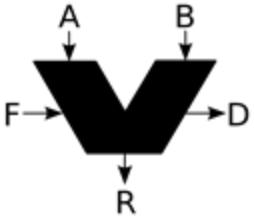


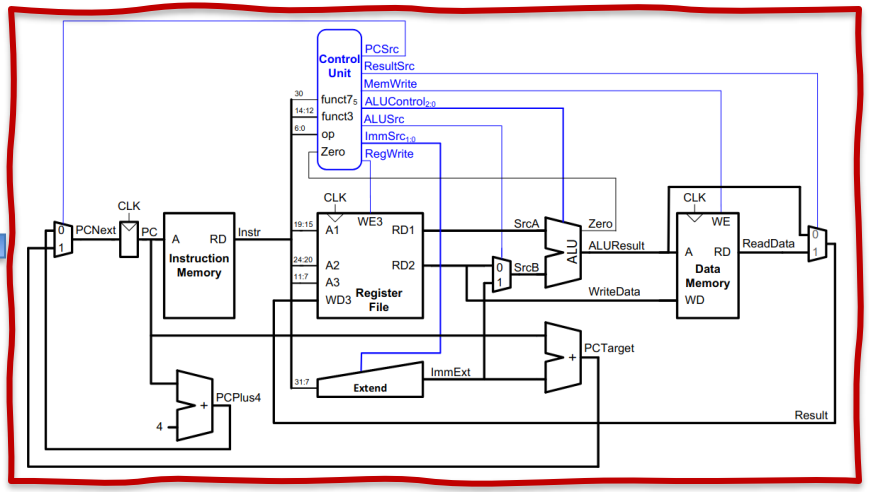
https://github.com/MKrekker/SINGLE-CYCLE-RISC-V

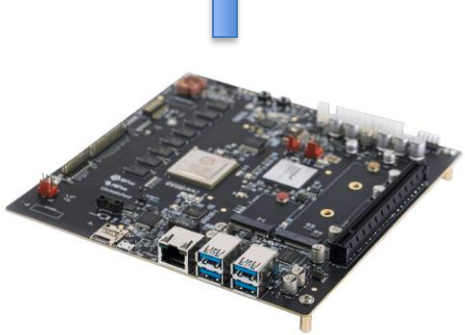
THE MAP (THE MACHINE)

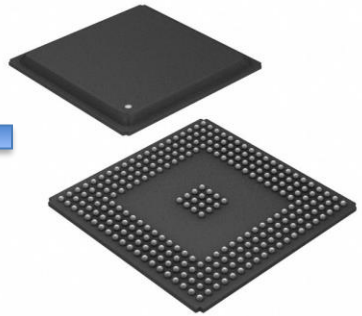






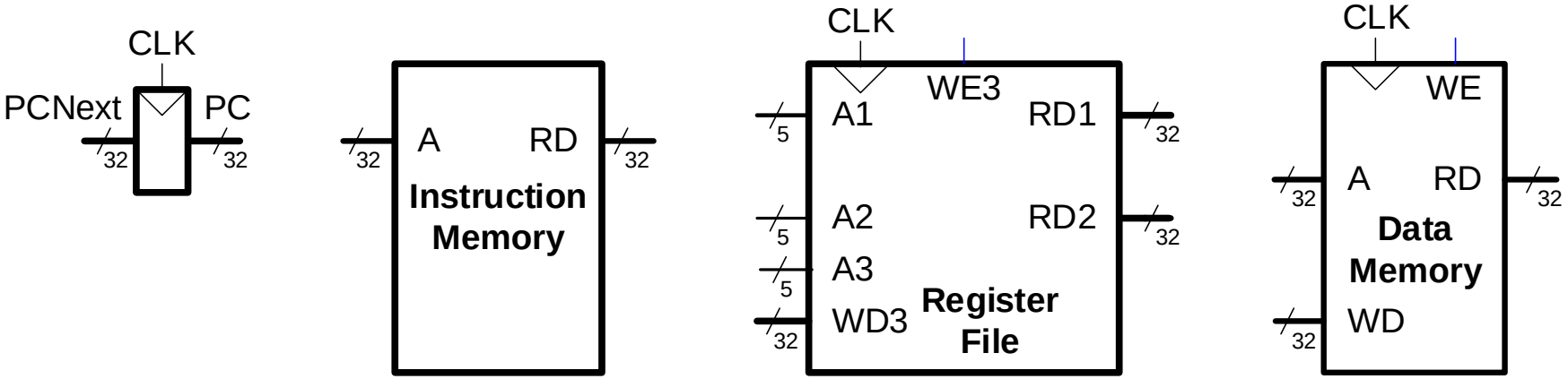




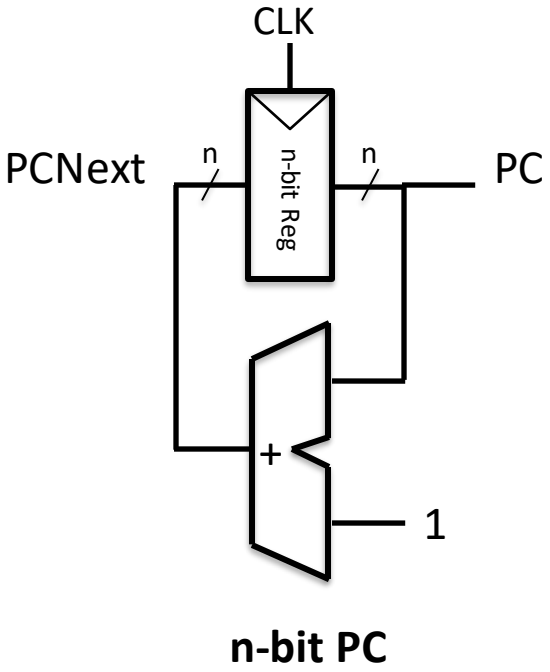


https://github.com/MKrekker/SINGLE-CYCLE-RISC-V

MEMORY COMPONENTS OF A PROCESSOR

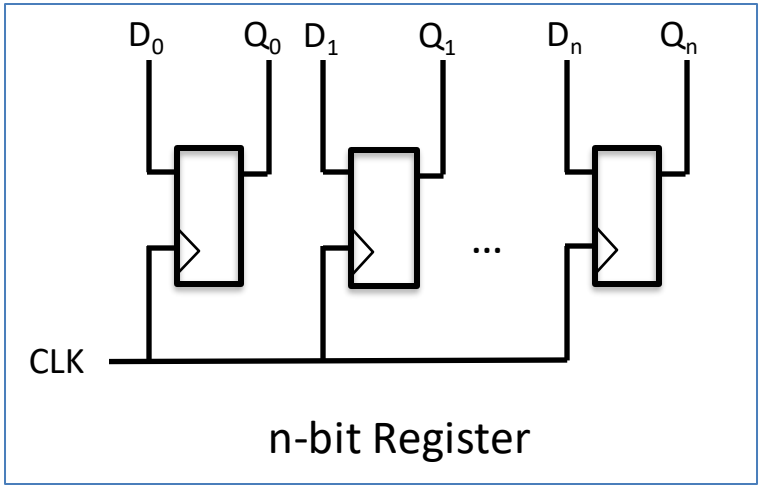


PROGRAM COUNTER



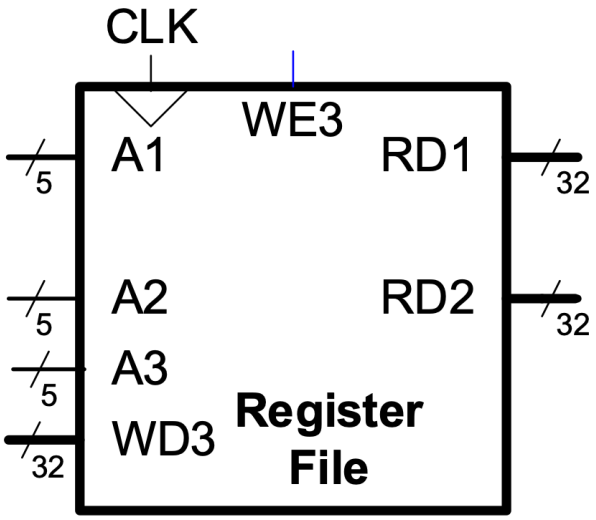
n-bit PC

- To track where we are in a program



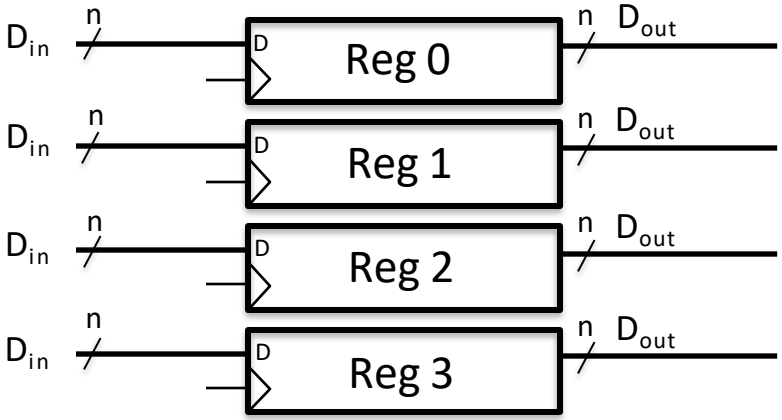
REGISTER FILE

- Temporary storage location
- Stores immediately needed variables
- External interface
  - Addresses: A1, A2, A3
  - Data: RD1, RD2, WD3
  - Enable: WE3
  - Clock: CLK

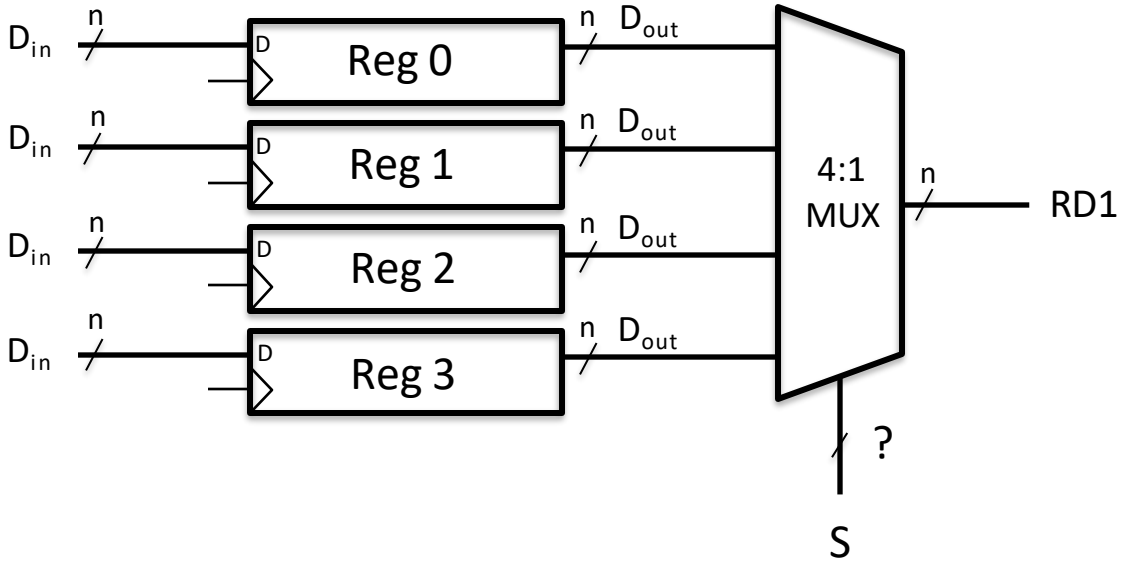


LET'S BUILD A REGISTER FILE OF FOUR REGISTERS

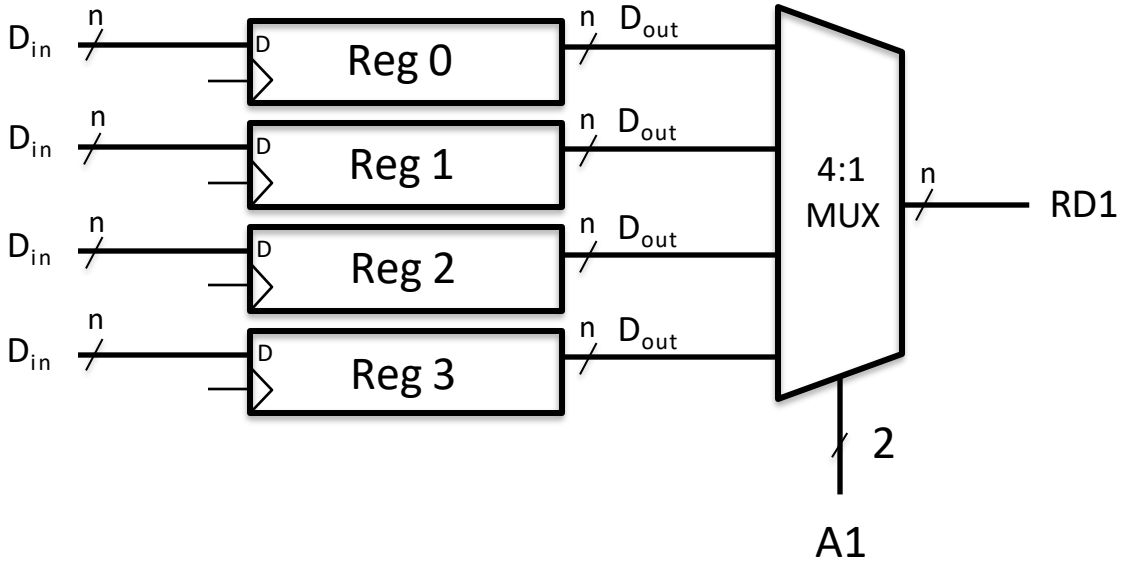
READ FROM A REGISTER FILE



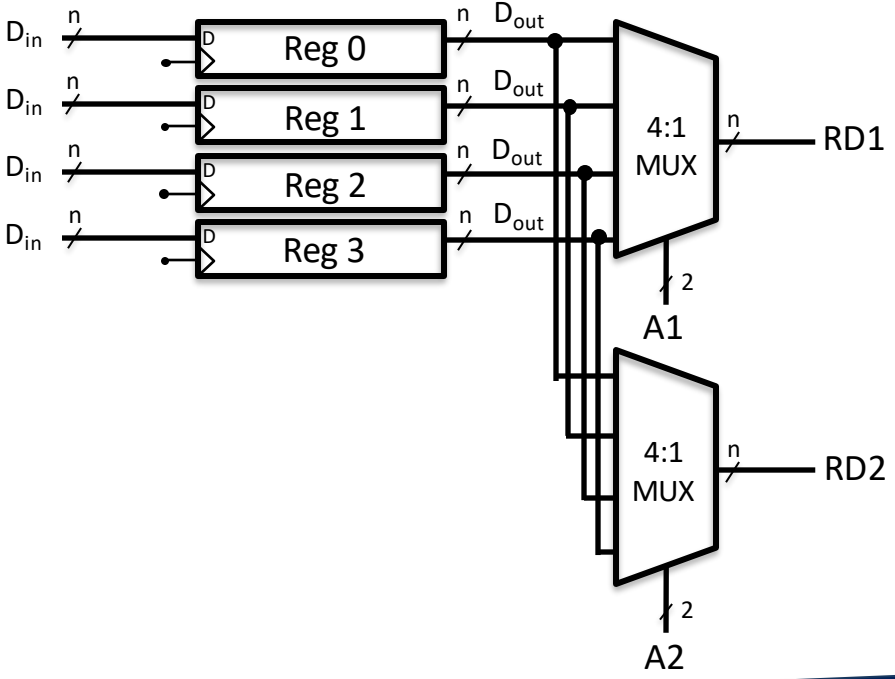
READ FROM A REGISTER FILE



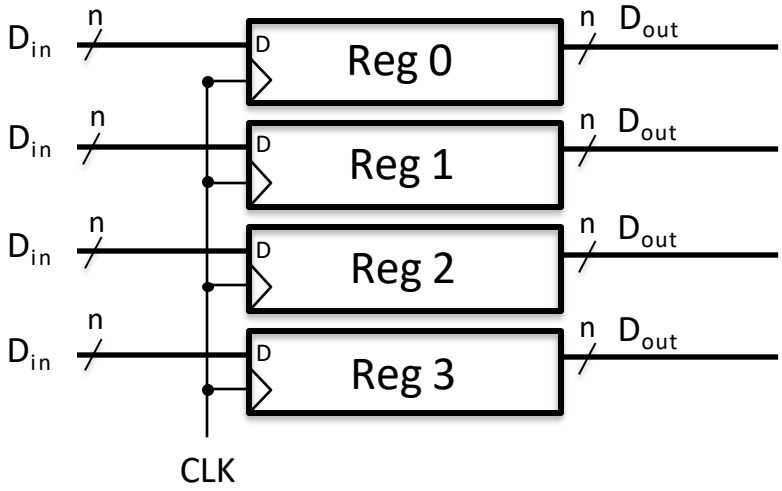
READ FROM A REGISTER FILE



READ FROM A REGISTER FILE

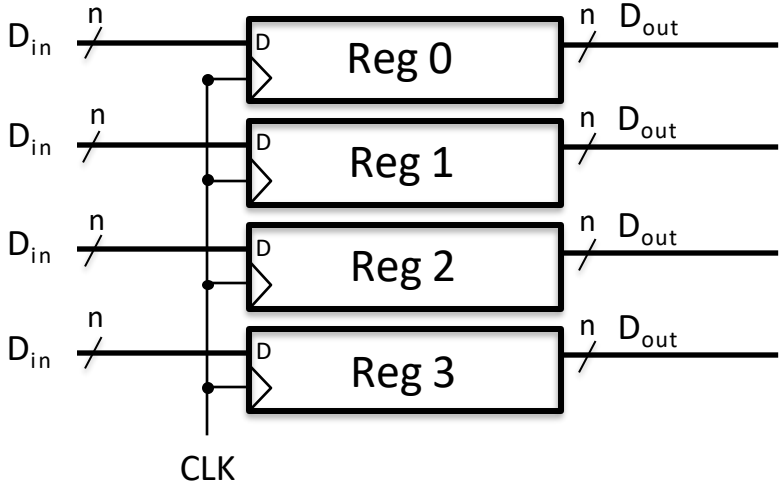


WRITE TO A REGISTER FILE



WRITE TO A REGISTER FILE

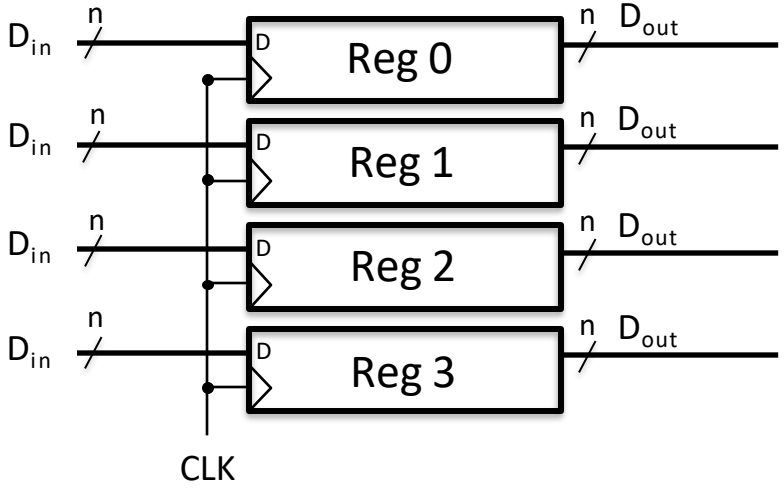
We want to write to a particular register when write enable is set.



WRITE TO A REGISTER FILE

We want to write to a particular register when write enable is set.

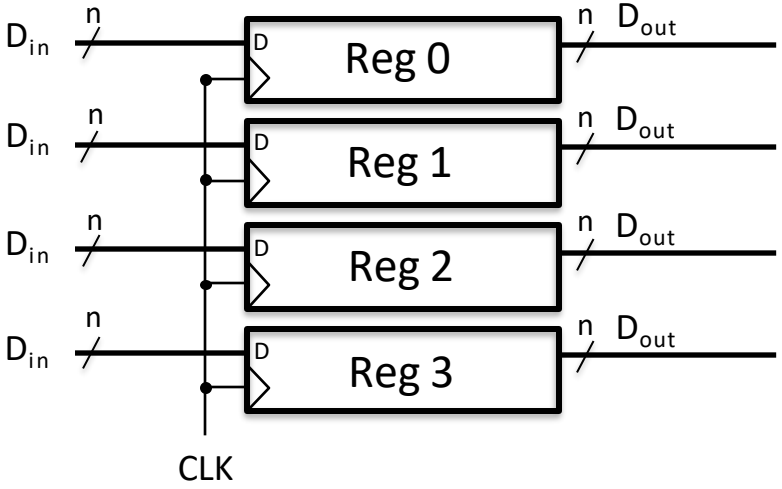
Do you see a problem with this circuit?



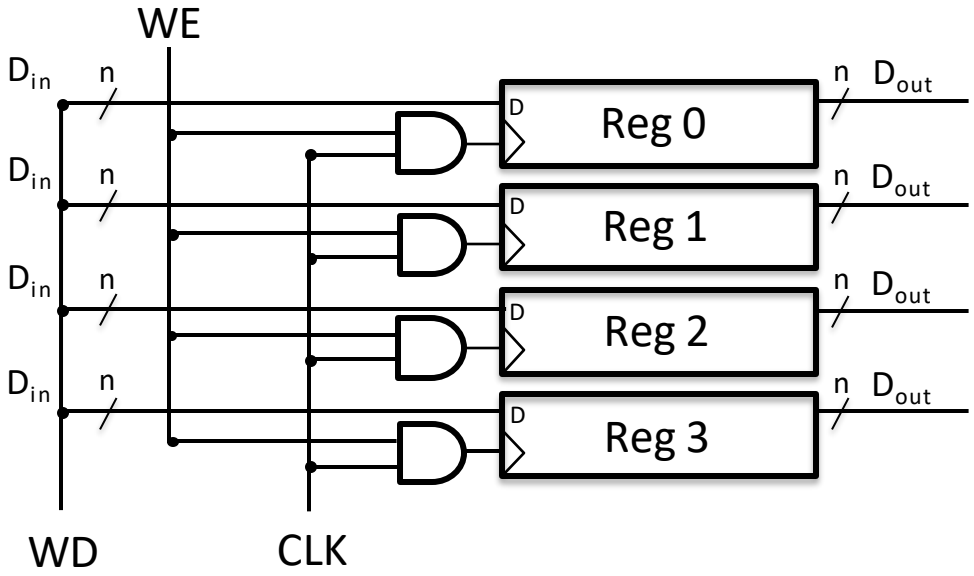
WRITE TO A REGISTER FILE

Additional input signals:

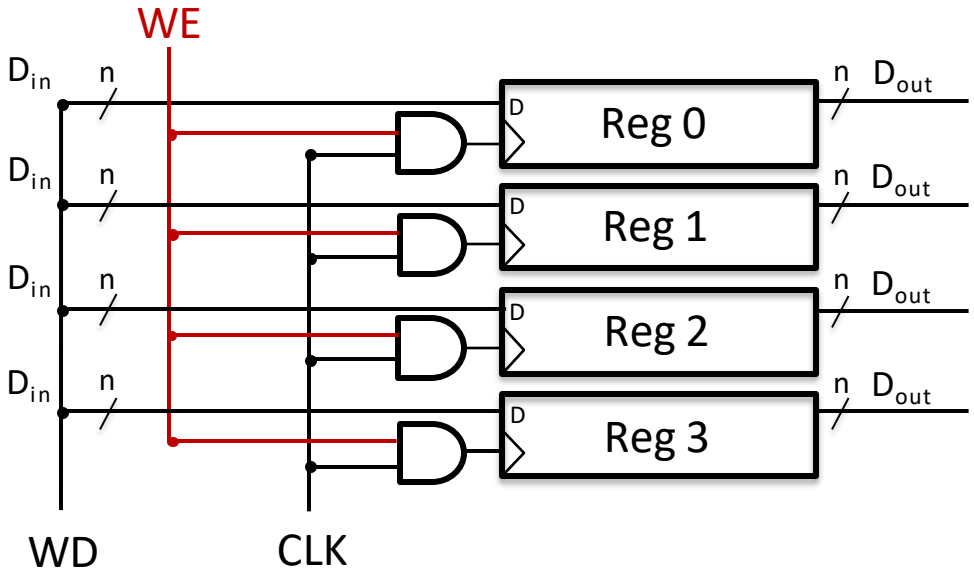
1. Write enable (WE)
2. Write data (WD)
3. Address of the register to be written (A3)



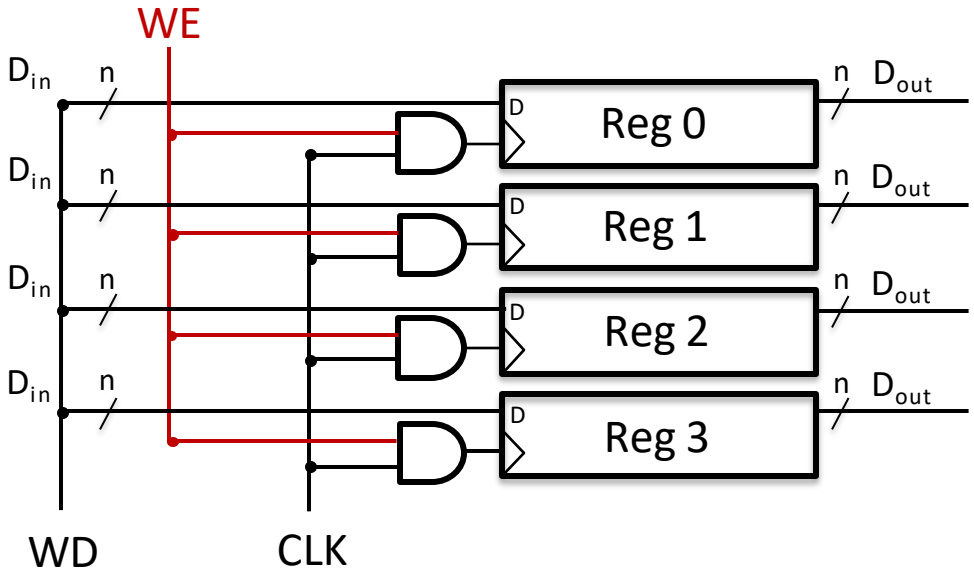
WRITE TO A REGISTER FILE



WRITE TO A REGISTER FILE



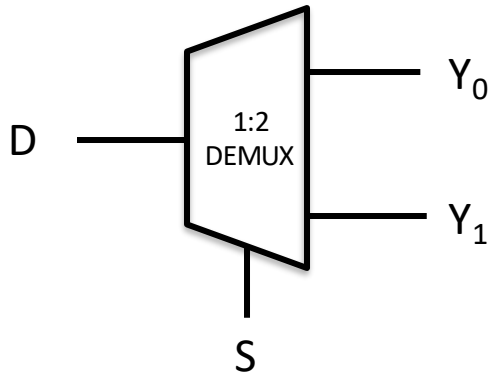
WRITE TO A REGISTER FILE



Are we missing a component?

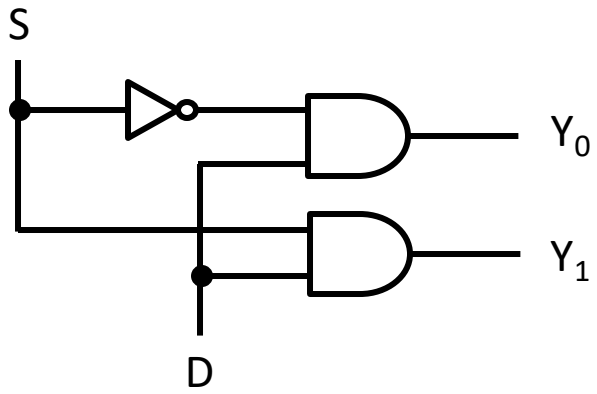
DEMULTIPLEXER (DEMUX)

Example: **1:2 DEMUX**

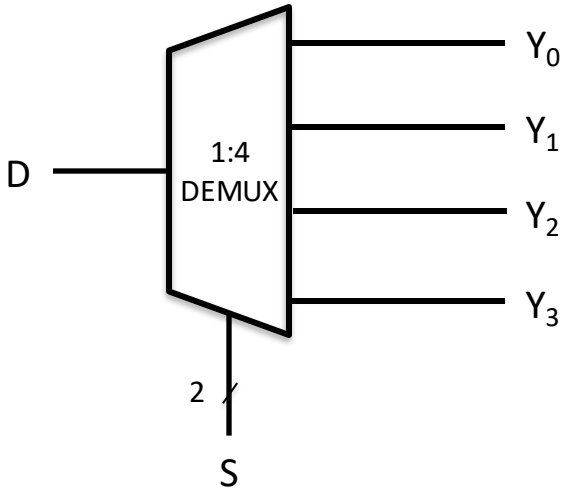


S	Y0	Y1
0	**D**	0
1	0	**D**

- Connects one input to one of the **N** outputs
- **Select** input is **$\log_2 N$ bits** – control input

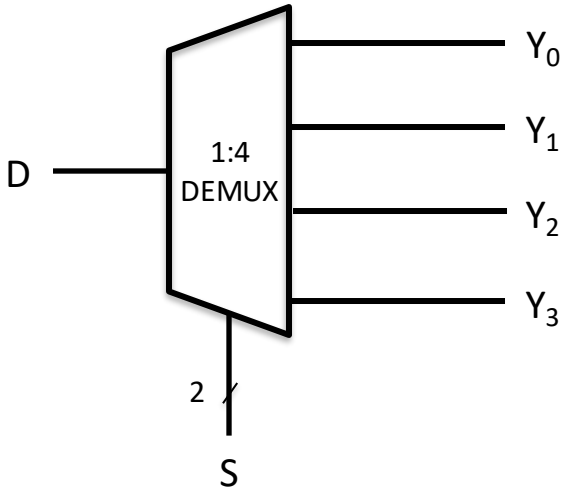


HIGH-ORDER DEMULTIPLEXER



S	Y0	Y1	Y2	Y3
00	**D**	0	0	0
01	0	**D**	0	0
10	0	0	**D**	0
11	0	0	0	**D**

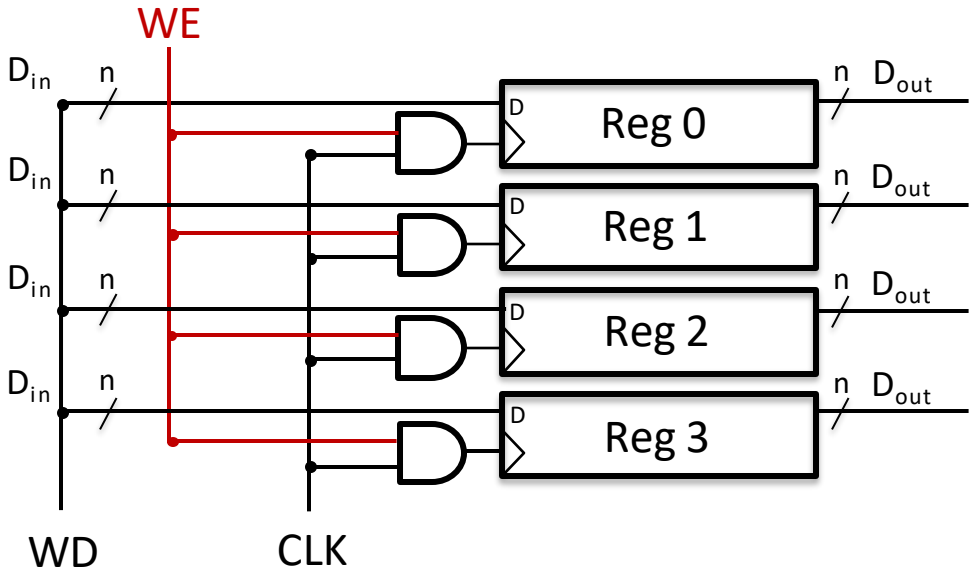
HIGH-ORDER DEMULTIPLEXER



S	Y0	Y1	Y2	Y3
00	**D**	0	0	0
01	0	**D**	0	0
10	0	0	**D**	0
11	0	0	0	**D**

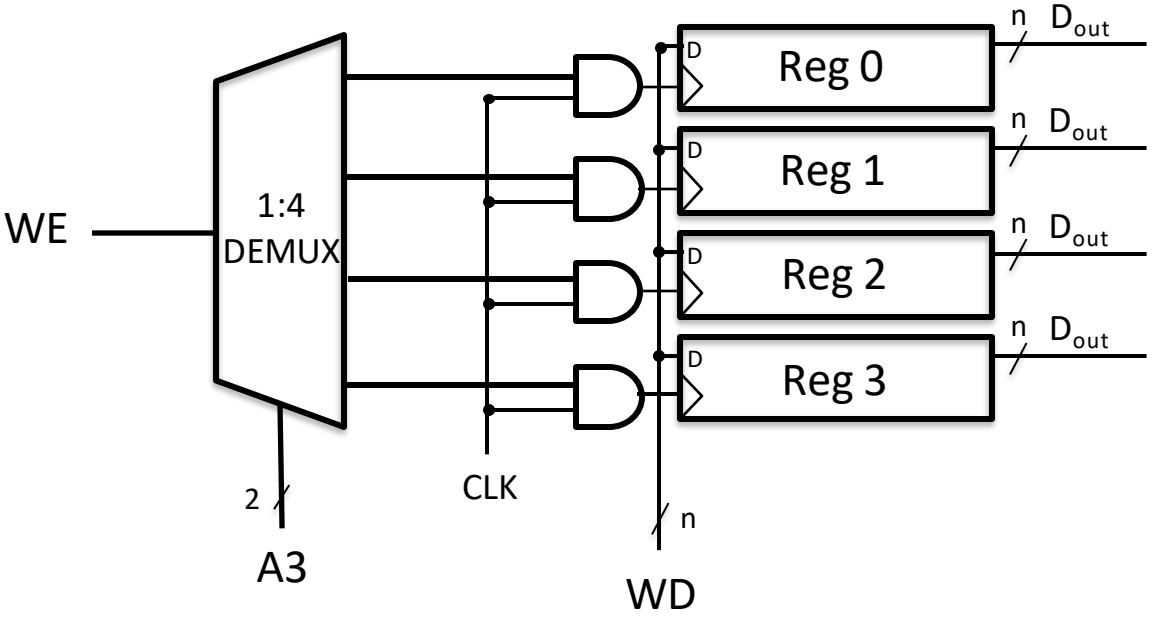
Can you implement higher-order demuxes like 1:8, 1:16, 1:64 using lower-order demuxes?

WRITE TO A REGISTER FILE



Are we missing a component?

WRITE TO A REGISTER FILE

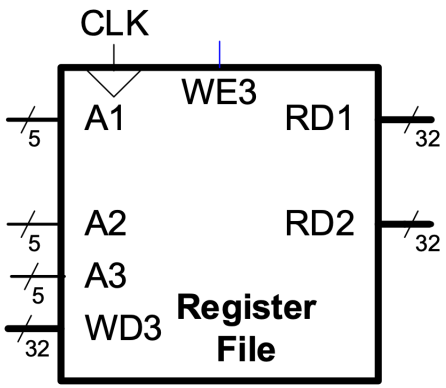


32-BIT REGISTER FILE

Simultaneously read from two registers and write into one register

Components:

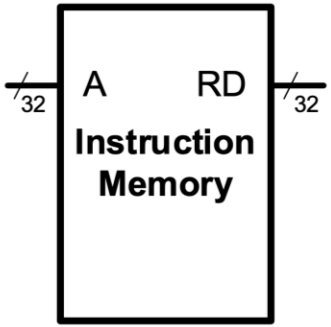
1. Multiplexers
2. Registers
3. Demultiplexers



INSTRUCTION MEMORY

- Stores the program

➢ Read data (RD) for a given address (A)

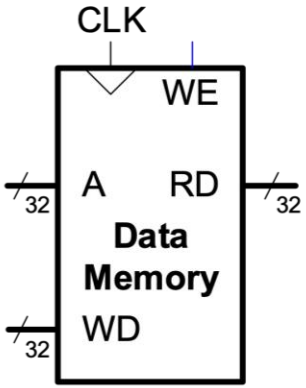


For this class, we will assume we cannot write to Instruction Memory.

DATA MEMORY

- Contains data needed by the program

- Read data (RD) from a given address (A)
- Write data (WD) to a given address (A)



EXERCISE

Here is an example of memory. Assume that the memory reads 4 bytes at a time and the values are stored in Big Endian.

```
000000C0 50 01 02 03 04 05 08 0D 15 22 37 46 FF AA C2 34
000000D0 3D 18 55 6D C2 2F F1 20 11 31 42 73 B5 28 DD 05
000000E0 E2 27 C9 B0 79 29 A2 CB 6D 38 A5 DD 82 5F E1 40
000000F0 21 72 83 E3 65 48 AD F4 A3 87 39 D0 09 DF E4 B5
```

EXERCISE

Here is an example of memory. Assume that the memory reads 4 bytes at a time and the values are stored in Big Endian.

```
000000C0 50 01 02 03 04 05 08 0D 15 22 37 46 FF AA C2 34
000000D0 3D 18 55 6D C2 2F F1 20 11 31 42 73 B5 28 DD 05
000000E0 E2 27 C9 B0 79 29 A2 CB 6D 38 A5 DD 82 5F E1 40
000000F0 21 72 83 E3 65 48 AD F4 A3 87 39 D0 09 DF E4 B5
```

- What will be the output when we read from address 0xD0?

EXERCISE

Here is an example of memory. Assume that the memory reads 4 bytes at a time and the values are stored in Big Endian.

```
000000C0 50 01 02 03 04 05 08 0D 15 22 37 46 FF AA C2 34
000000D0 3D 18 55 6D C2 2F F1 20 11 31 42 73 B5 28 DD 05
000000E0 E2 27 C9 B0 79 29 A2 CB 6D 38 A5 DD 82 5F E1 40
000000F0 21 72 83 E3 65 48 AD F4 A3 87 39 D0 09 DF E4 B5
```

- What will be the output when we read from address 0xD0?

Answer: 0x3D18556D

EXERCISE

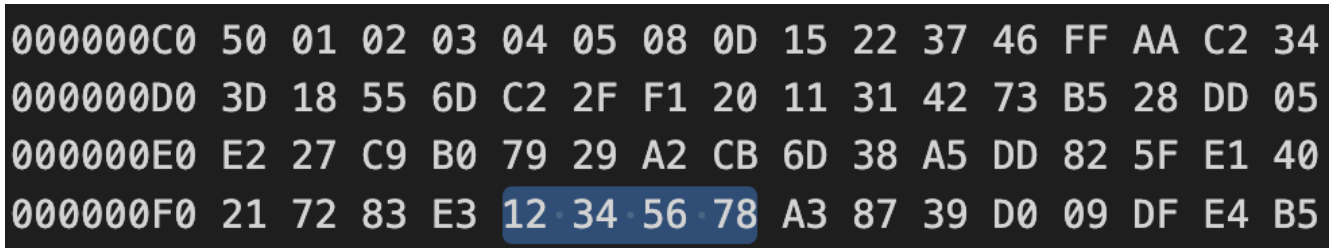
Here is an example of memory. Assume that the memory reads 4 bytes at a time and the values are stored in Big Endian.

```
000000C0 50 01 02 03 04 05 08 0D 15 22 37 46 FF AA C2 34
000000D0 3D 18 55 6D C2 2F F1 20 11 31 42 73 B5 28 DD 05
000000E0 E2 27 C9 B0 79 29 A2 CB 6D 38 A5 DD 82 5F E1 40
000000F0 21 72 83 E3 65 48 AD F4 A3 87 39 D0 09 DF E4 B5
```

- How does the memory change if we write 0x12345678 to memory address 0xF4?

EXERCISE

Answer:



EXERCISE

```
000000C0 50 01 02 03 04 05 08 0D 15 22 37 46 FF AA C2 34
000000D0 3D 18 55 6D C2 2F F1 20 11 31 42 73 B5 28 DD 05
000000E0 E2 27 C9 B0 79 29 A2 CB 6D 38 A5 DD 82 5F E1 40
000000F0 21 72 83 E3 12 34 56 78 A3 87 39 D0 09 DF E4 B5
```

- What is the size of this memory?

EXERCISE

```
000000C0 50 01 02 03 04 05 08 0D 15 22 37 46 FF AA C2 34
000000D0 3D 18 55 6D C2 2F F1 20 11 31 42 73 B5 28 DD 05
000000E0 E2 27 C9 B0 79 29 A2 CB 6D 38 A5 DD 82 5F E1 40
000000F0 21 72 83 E3 12 34 56 78 A3 87 39 D0 09 DF E4 B5
```

- What is the size of this memory?

Answer: 64 bytes

IS THIS IT?

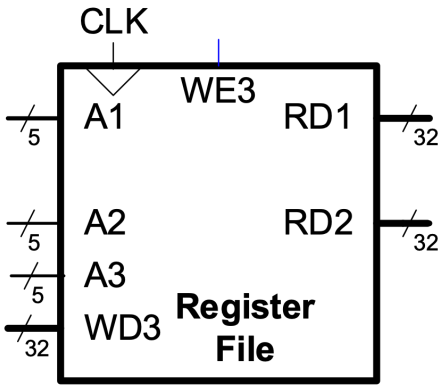
- Are the RAMs in your laptop just made of flipflops?
- Are the hard disks in your computer systems just made of flipflops?
- Do you have other memory components in your computer?

EXERCISE

Assume register counting starts from 0.

What should be the input signals to

1. write 0xABCD to Register #8?
2. read from Register #31 and Register #16?

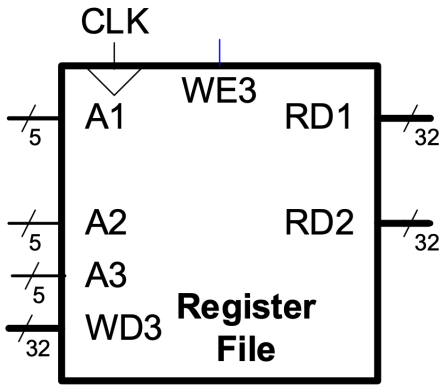


EXERCISE

Assume register counting starts from 0.

What should be the input signals to

1. write 0xABCD to Register #8?
2. read from Register #31 and Register #16?



Answer:

1. WE3 : 1, WD3 : 0xABCD, A3 : 0x08
2. A1: 0x1F, A2: 0X10

QUESTIONS?