
CSE 220 Computer Organization

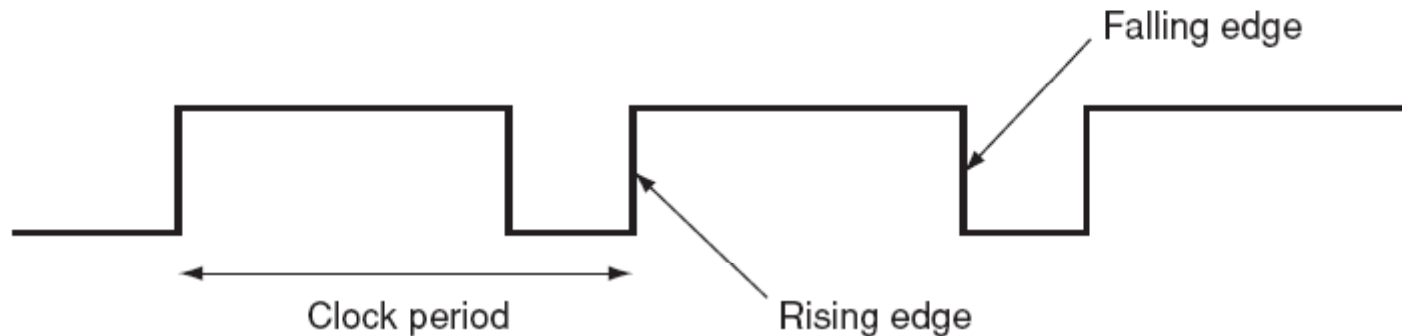
Lecture 18

Hussein Badr
Computer Science, Stony Brook University

<http://www.cs.sunysb.edu/~cse220>

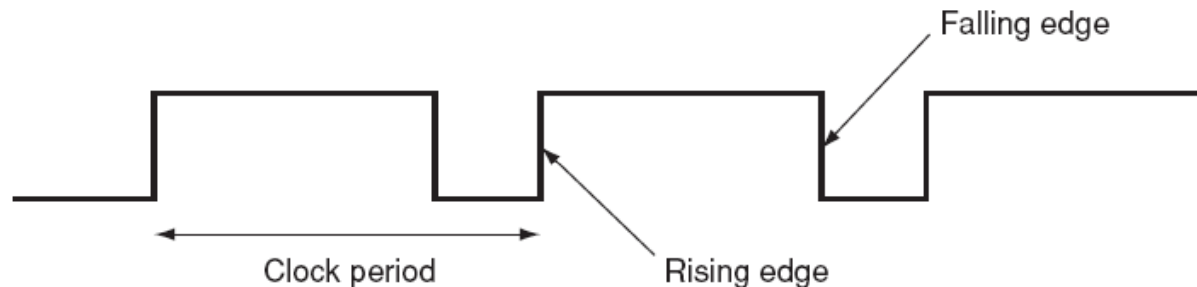
Digital Clock (1/2)

- The clock period, or cycle time, is the length of time of the critical path in the circuit.
- The clock cycle time or clock period is divided into two portions:
 - when the clock is high (1)
 - when the clock is low (0)



Digital Clock (2/2)

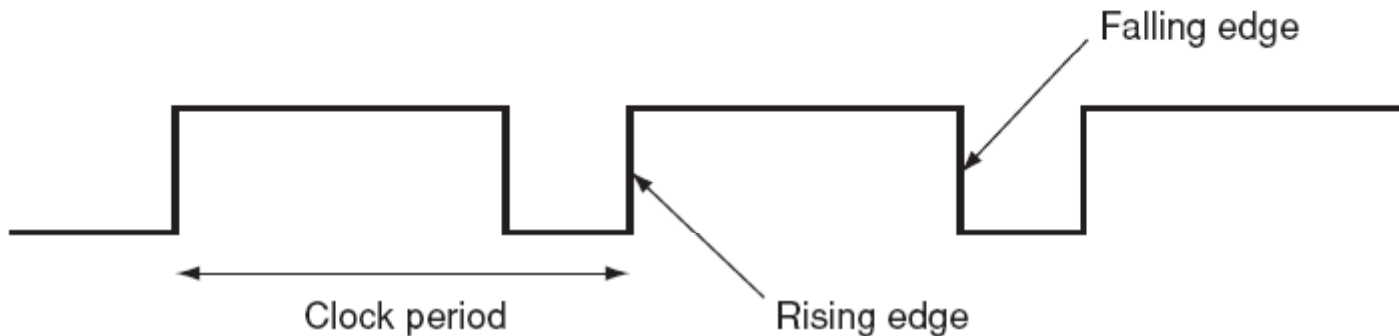
- The unit is in seconds. Clock speed is the number cycles per second. *E.g.*, 500 MHz clock has 500 million cycles per second. Its cycle time is $1 / (500 * 10^6) = 2$ nanoseconds.
- Clock speed or *frequency* is quoted in KHz, MHz, or GHz.



- Usually the faster the clock, the faster (the more work) the CPU does.
 - Some work is performed every cycle of the clock.
 - The faster the clock the more work done per second.

How are clocks used?

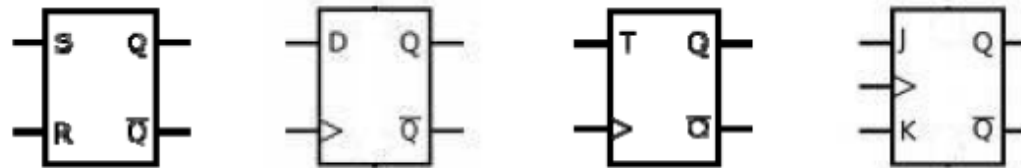
- **Clocks are needed to decide when an element that contains a state should be updated**
- **State elements:**
 - a memory element that gets updated on a clock edge



Flip-Flops (1/2)

- **Flip-Flops are a storage/memory element which stores one bit (binary digit) of data.**
- **A flip-flop is controlled by (usually) one or two control signals and/or a gate or clock signal. The output often includes the complement as well as the normal output.**
- **There are multiple types of Flip Flops**
 - **Set-Reset (S-R) FF**
 - **Delay (D) FF (built from S-R)**
 - **Toggle (T) FF**
 - **J-K FF (built from S-R)**
- **Any one of these flip-flop types can be used to build any of the others.**
- **Flip flops are used to save information from one clock cycle to the next.**

Flip-Flops (2/2)

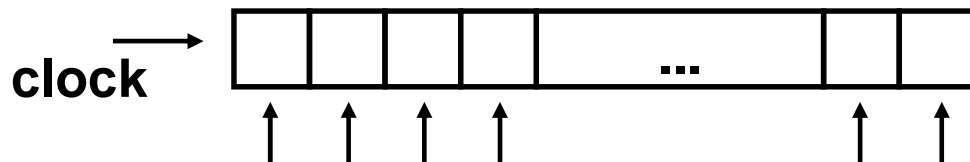


'>' denotes the clock input

S-R Flip-Flop			Delay Flip-Flop				Toggle Flip-Flop				J-K Flip-Flop			
							$Q_{next} = T \oplus Q$							
S	R	Action	Clock	D	Q	Q_{prev}	T	Q	Q_{next}	Comment	J	K	Q_{next}	Comment
0	0	Keep state	Rising edge	0	0	X	0	0	0	hold state	0	0	Q_{prev}	hold state
0	1	Q = 0	Rising edge	1	1	X	0	1	1	hold state	0	1	0	reset
1	0	Q = 1	Non-Rising	X	constant		1	0	1	toggle	1	0	1	set
1	1	Unstable combination, (race condition)					1	1	0	toggle	1	1	$\overline{Q_{prev}}$	toggle
			'X' denotes a 'Don't Care' condition, meaning the signal is irrelevant.											
			The Q output always takes on the state of the D input at the moment of a rising clock edge, and never at any other time. It is called the D flip-flop for this reason, since the output takes the value of the D input or Data input, and Delays it by one				If the T input is high, the T flip-flop changes state ("toggles") whenever the clock input is strobed. If the T input is low, the flip-flop holds the previous value.				Augments the behavior of the S-R flip-flop by interpreting the S = R = 1 condition as a "flip" or toggle command.			

Register (1/2)

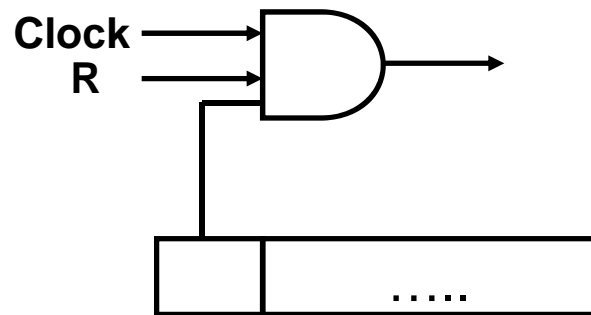
- A register is an array of D flip-flops.
- A 32-bit register has 32 D flip-flops.
- How do we read a register? The 32 Q output bits supply the value.
- How do we write to a register?



- The value to be stored is at the 32 D inputs. Then we clock the register just once. So whatever number is present at the inputs is loaded in.
- We clock the register only when we want to write.

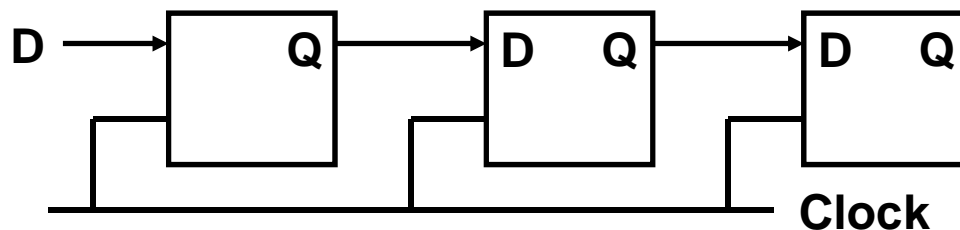
Register (2/2)

- **How to read in synchronization with the clock?**



When 'read bit' $R = 1$ and clock is 1, you read the bits and they are available for use.

- **Shift register: If D flip flops are connected in series, it acts as a shift register.**



- **Every time we clock, data will move from left to right one position.**

Register File (1/3)

- **Central to MIPS datapath.**
- **It is a set of registers.**
- **We read or write a register by supplying its number.**
- **This example has 2 read ports.**
It can read two registers at a time.
- **It has one write port.**
- **An R-type instruction has two source operands and one destination.**

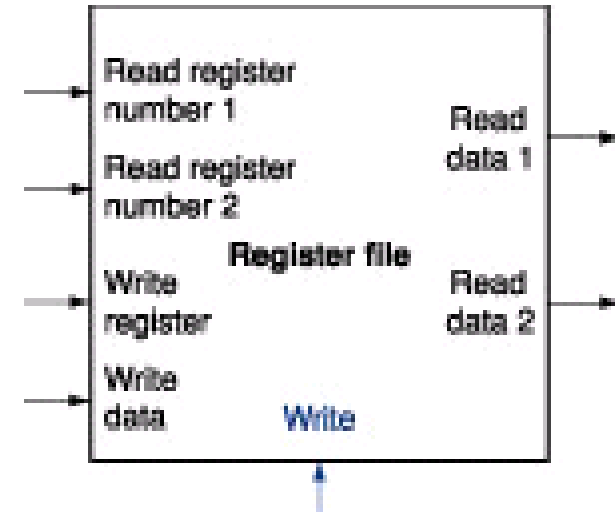


FIGURE C.8.7 A register file with two read ports and one write port has five inputs and two outputs. The control input Write is shown in color.

Register File (2/3)

- Read ports are implemented using a multiplexor.
- We supply a number to read each register.
- The number (5 bits wide) controls a multiplexor.

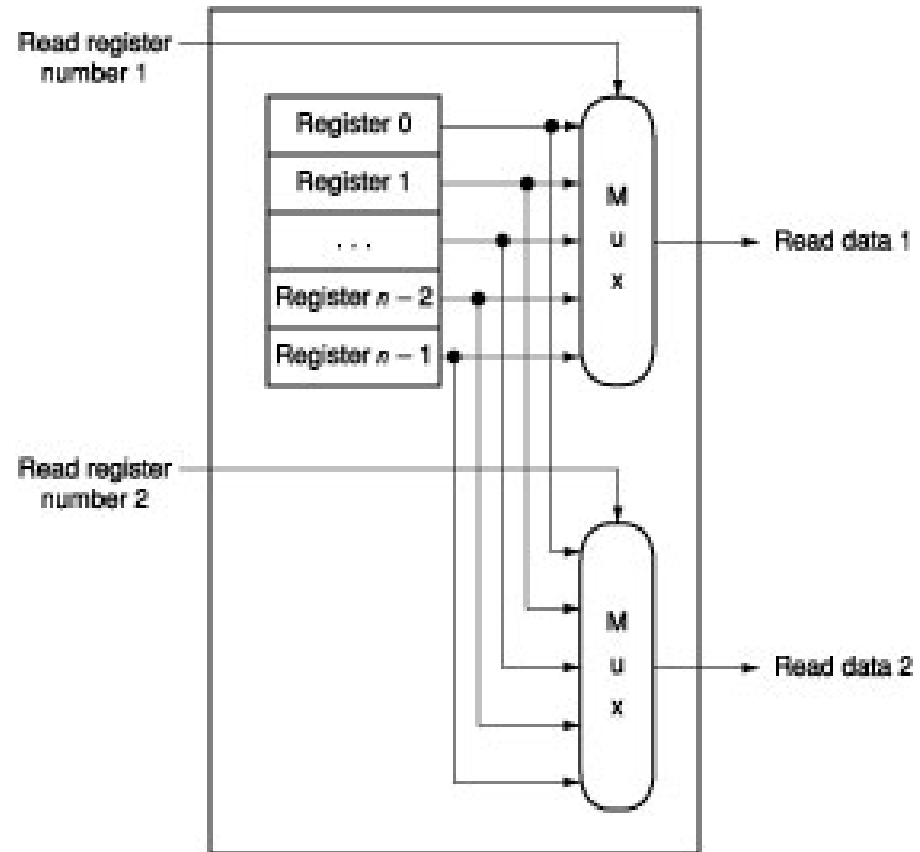


FIGURE C.8.8 The implementation of two read ports for a register file with n registers can be done with a pair of n -to-1 multiplexors each 32 bits wide. The register read number signal is used as the multiplexor selector signal. Figure C.8.9 shows how the write port is implemented.

Register File (3/3)

- When writing, we usually change the contents of only one register.
- Decode the supplied register number to select the designated register.
- Data to be written is sent to the D inputs of all registers.
- But only the selected register is 'clocked'.

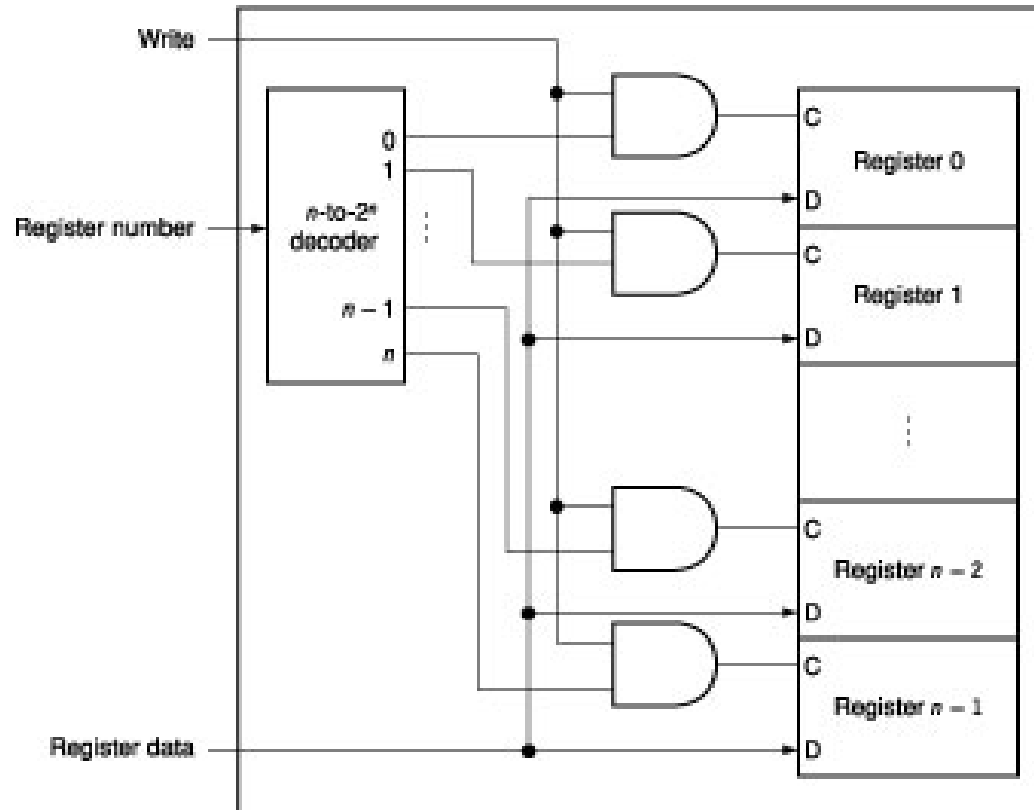


FIGURE C.8.9 The write port for a register file is implemented with a decoder that is used with the write signal to generate the C input to the registers. All three inputs (the register number, the data, and the write signal) will have setup and hold-time constraints that ensure that the correct data is written into the register file.