CMPT 295

Unit – Microprocessor Design & Instruction Execution

Lecture 27 – Combinational and Sequential Logic Circuits

Last Lecture

- We have now started to explore how the microprocessor executes machine instructions (series of 0's and 1's)
  - More specifically, how its datapath can be constructed
- Microprocessor itself is …
  - Made of resistors, capacitors, diodes, and transistors
  - Billions of them, so understanding how these small components work together is too detailed hence too onerous
  - So we resort to abstraction (***black box***) in order to understand their functioning
    - We use **logic gates** to abstract the "how these small components work together" by highlighting "what they do", i.e., perform a Boolean function
  - What is important for us to understand is that hardware components (such as logic gates) have propagation delay
    - Signals (0's and 1's) take time to propagate through them

Today's Menu

- Instruction Set Architecture (ISA)
  - Definition of ISA
- Instruction Set design
  - Design principles
  - Look at an example of an instruction set: MIPS
  - Create our own
  - ISA evaluation
- Implementation of a microprocessor (CPU) based on an ISA
  - Execution of machine instructions (datapath)
  - Intro to logic design + Combinational logic + Sequential logic circuit
  - Sequential execution of machine instructions
  - Pipelined execution of machine instructions + Hazards

From our last lecture - Digital circuits

Types of components found in a digital system such as a microprocessor

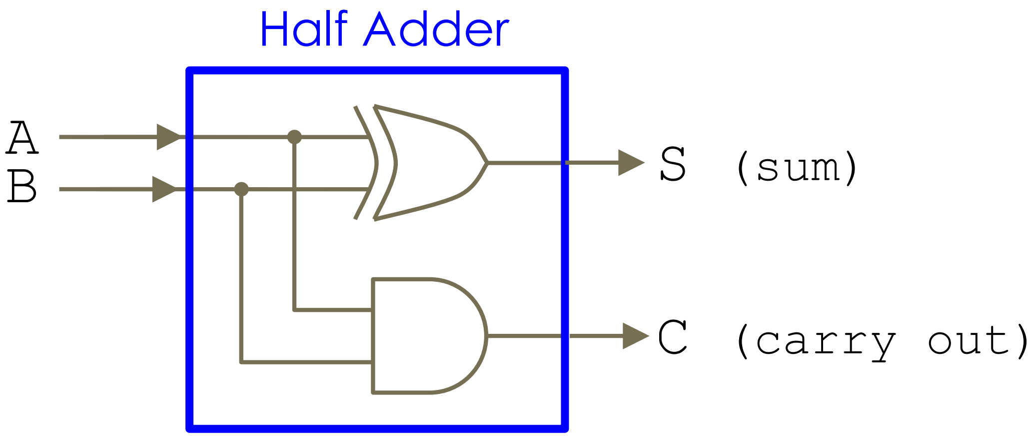
- In order to understand how the microprocessor executes these machine instructions (series of 0's and 1's), we need to have a look at the components of a microprocessor and how they function:
  1. Combinational logic -> manipulate bits (compute functions on bits e.g., ADD)
  2. Memory elements -> store bits
  3. Clock signals -> regulate the update of memory elements

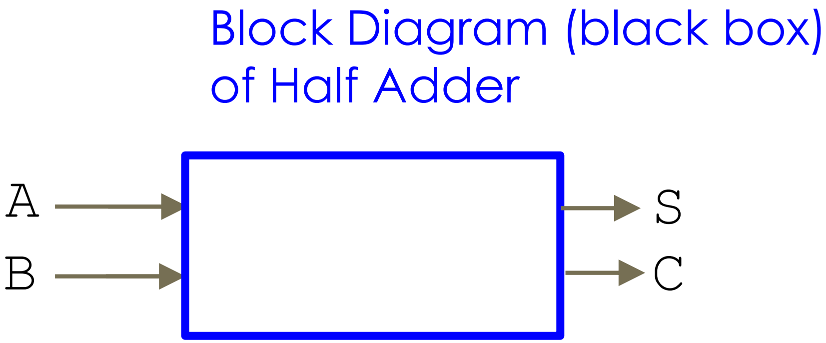
  and what affects the execution speed of these components such as propagation delay
- So, we need to understand a few things about ***digital circuits***

Combinational logic circuits - 1

•output depends only on input
•given same input produces same output
•no internal storage

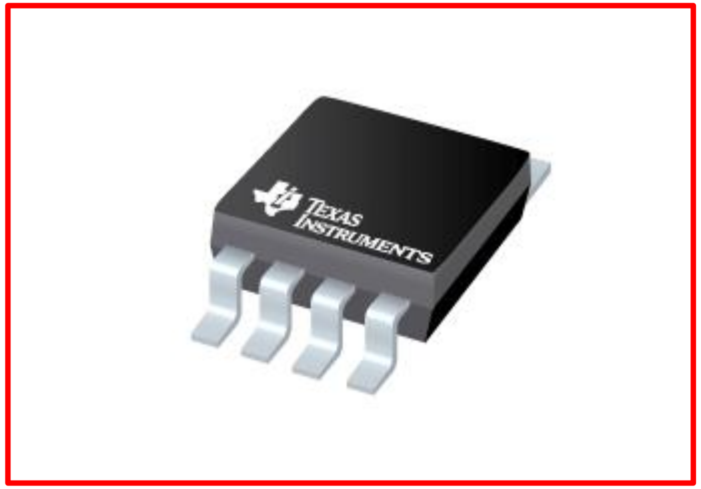
- Made by connecting several logic gates together
- Compute more complex functions than just AND or XOR
- Example: Combinational logic circuit that adds 2 bits together

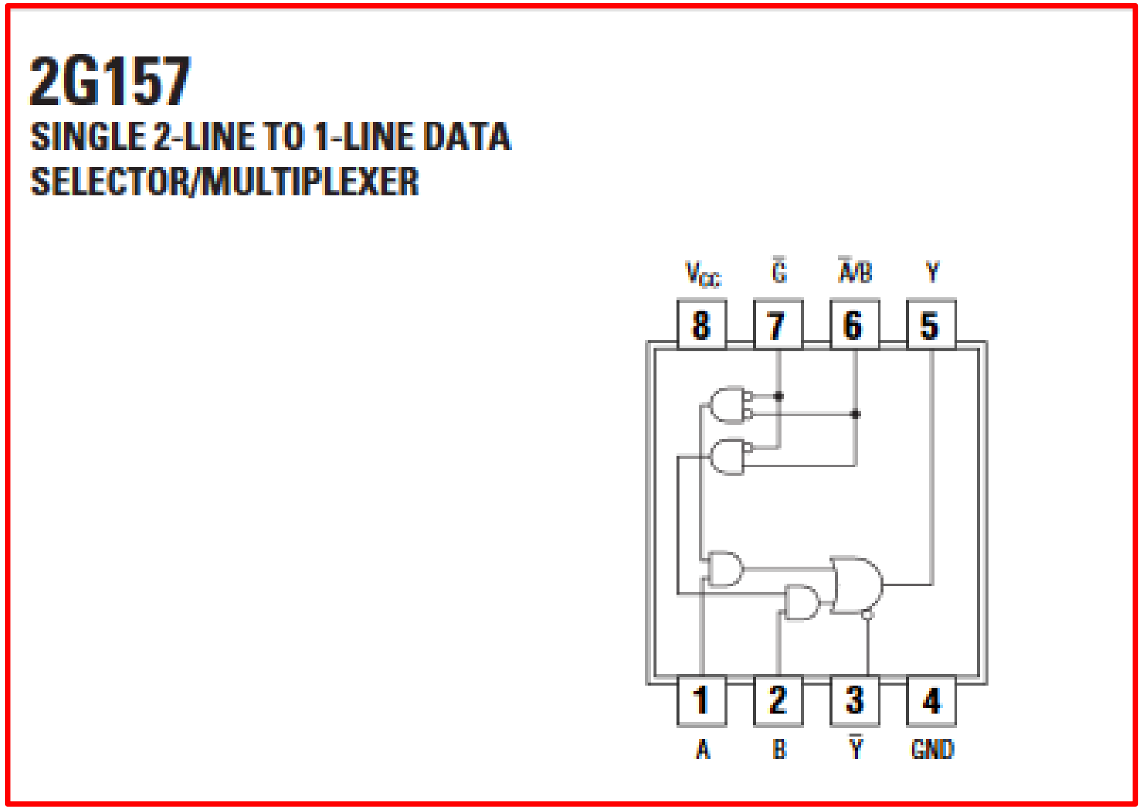




- Propagation delay: "1 gate" delay
- http://sullystationtechnologies.com/ichalfadder.html

Example of an integrated circuit (IC) **SN74LVC2G157**

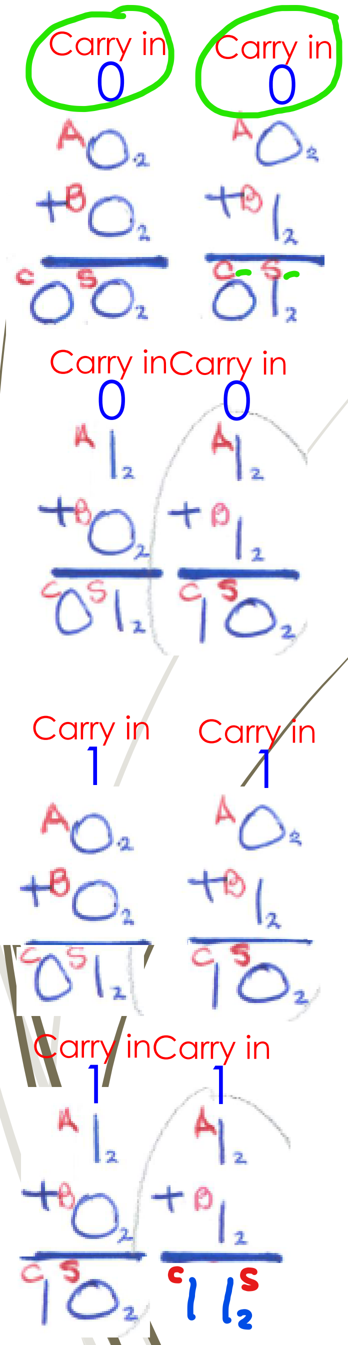




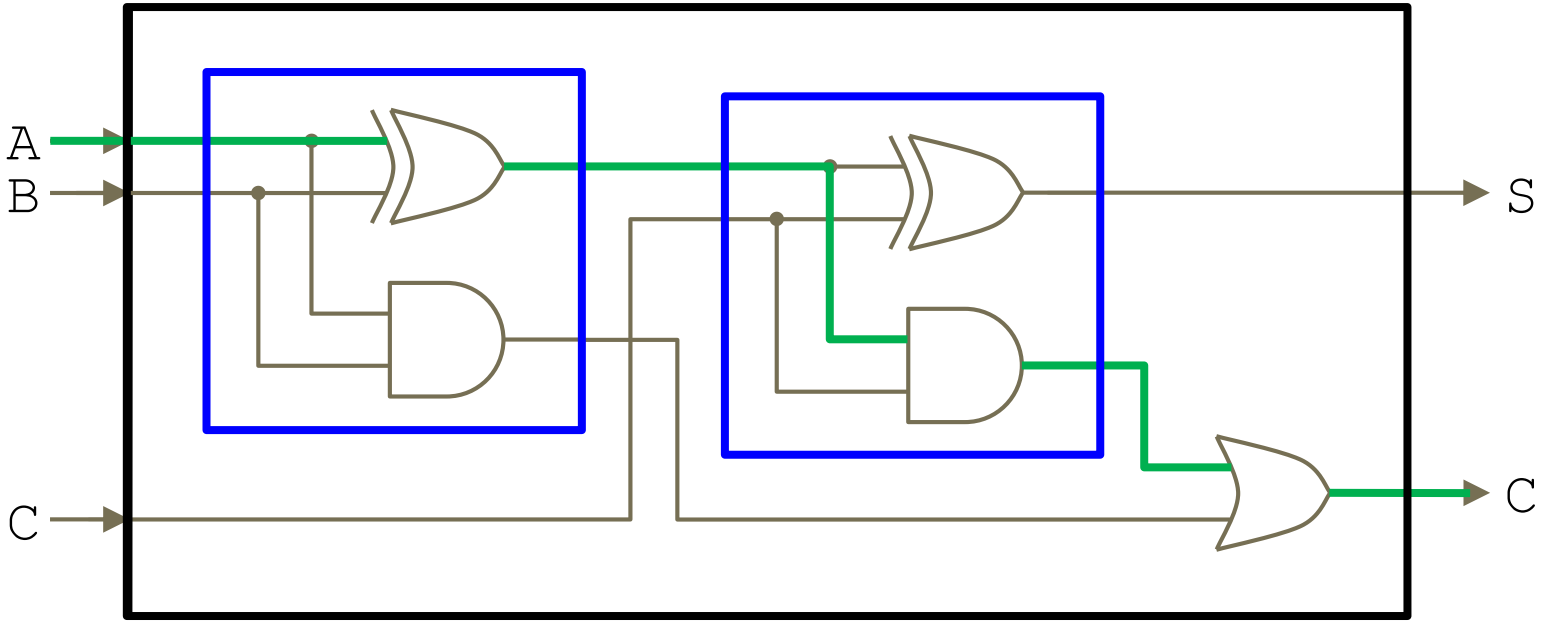
More examples in our Textbook

- Combinational circuit to test for bit equality -> Figure 4.10
- Combinational circuit to test for word equality -> Figure 4.12
- Single-bit multiplexor circuit -> Figure 4.11
- Word-level multiplexor circuit -> Figure 4.13

Combinational logic circuits - 2



- Can connect several combinational logic circuits together to perform more complex functions and/or to perform function on a wider input
- What does this circuit do?



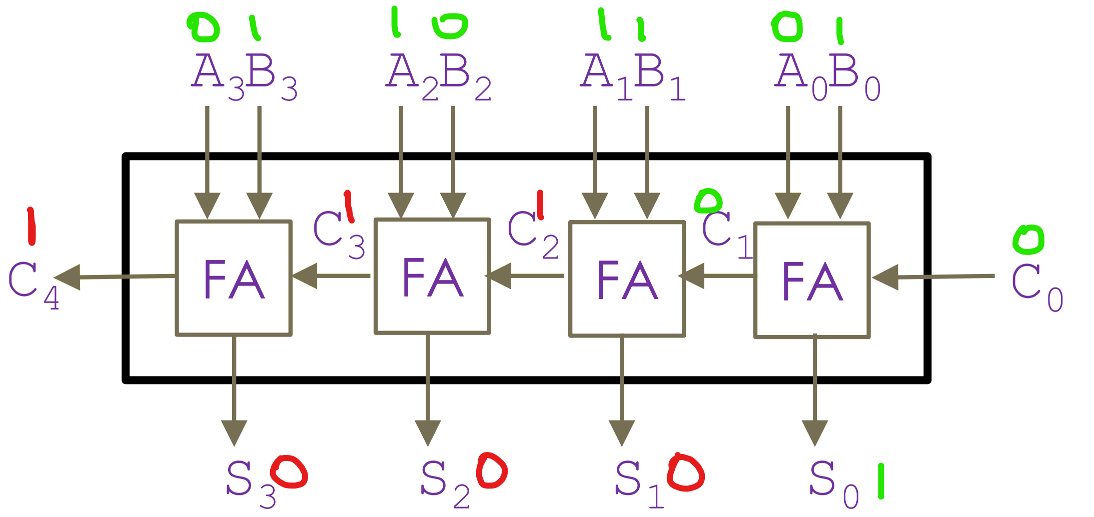
- Propagation delay: “3 gate” delay

Hw: label all input & output lines!

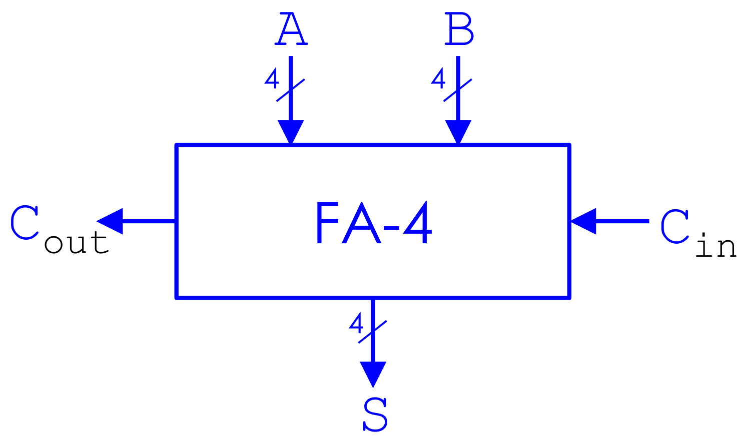
Combinational logic circuits - 3

- Can connect several combinational logic circuits together as **black boxes** (black box design can be modular)
- The resulting circuit can operate on word size data such as integral values and memory addresses
- Example: Full Adder operating on data of word size 4 (w = 4)

aka 4-bit ripple carry adder



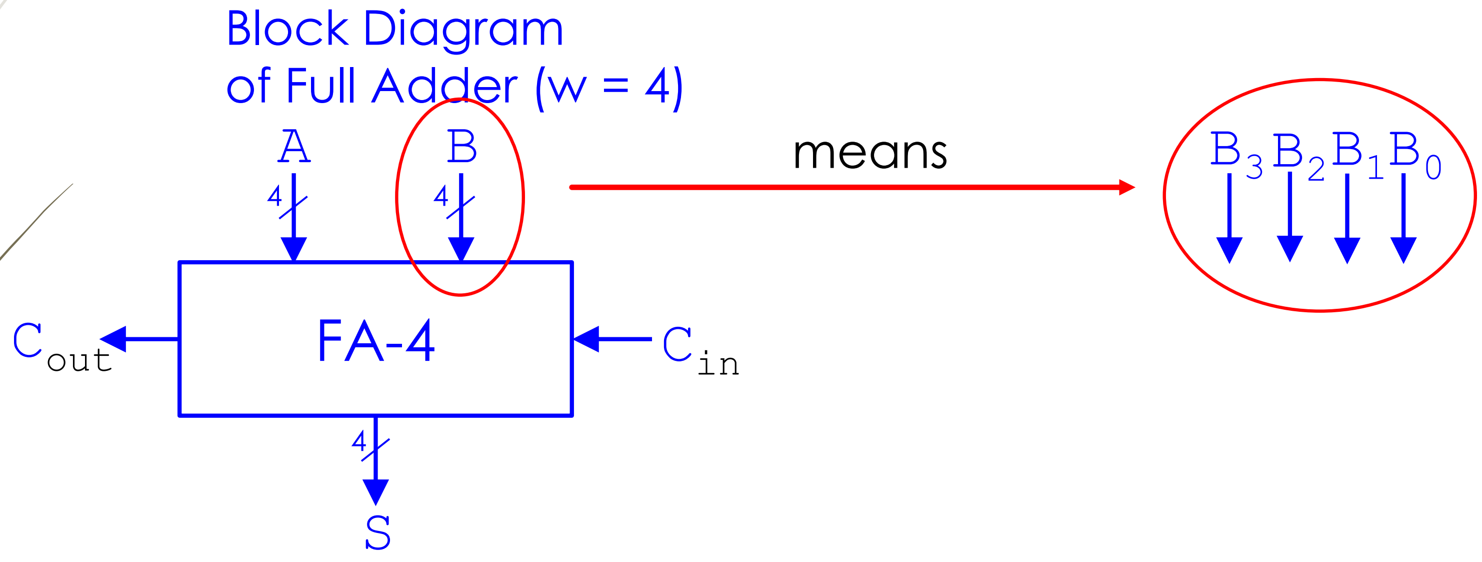
Block Diagram of Full Adder (w = 4)



C_3 C_2 C_1 C_0
1 1 0 0
A 0 1 1 0
+B 1 0 1 1
S_3 S_2 S_1 S_0
1 0 0 0 1
C_4

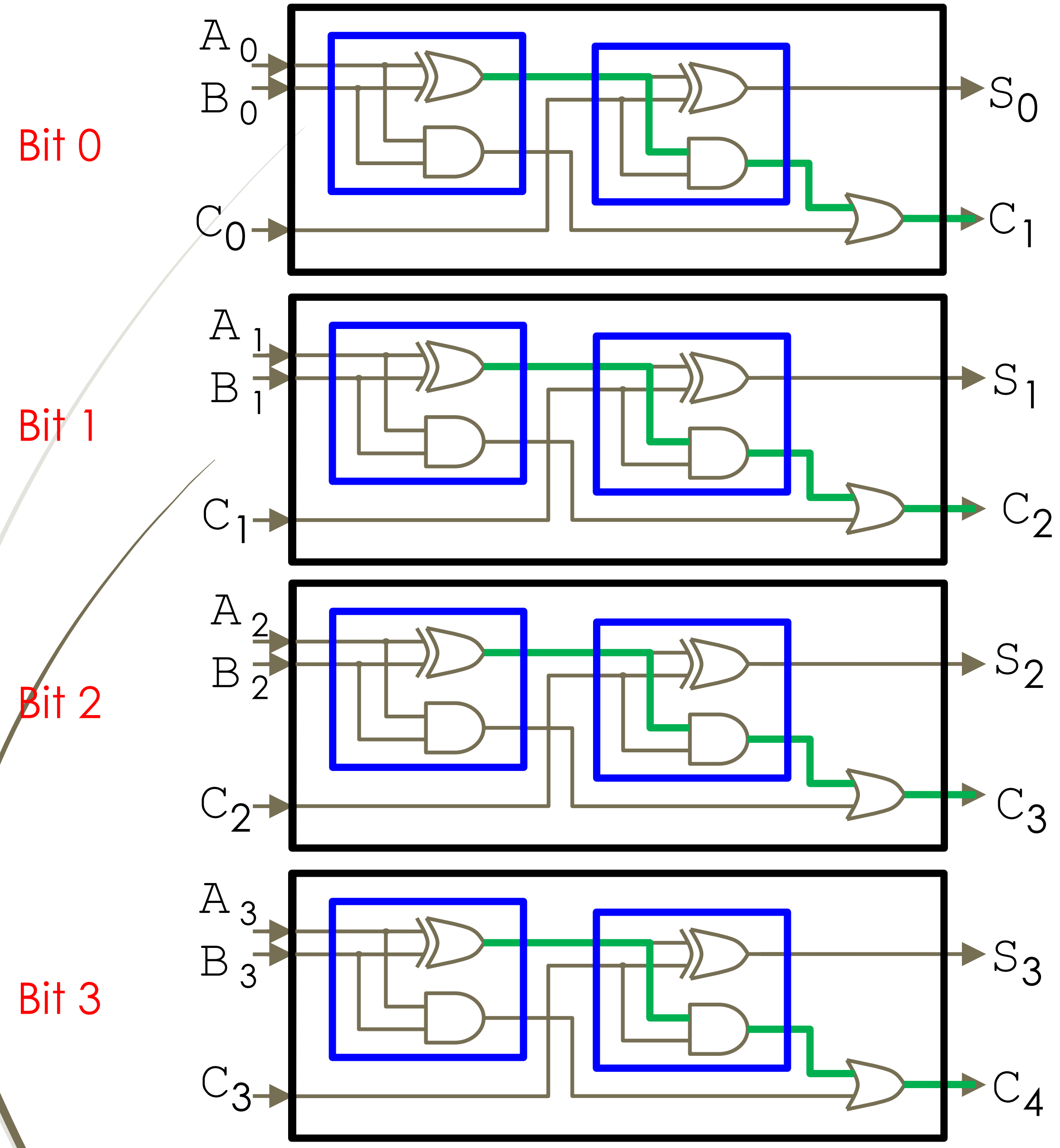
- Propagation delay?

where …



Block Diagram
of Full Adder (w = 4)
A
B
4
4
FA-4
Cout
Cin
4
S
means
B3 B2 B1 B0

Homework

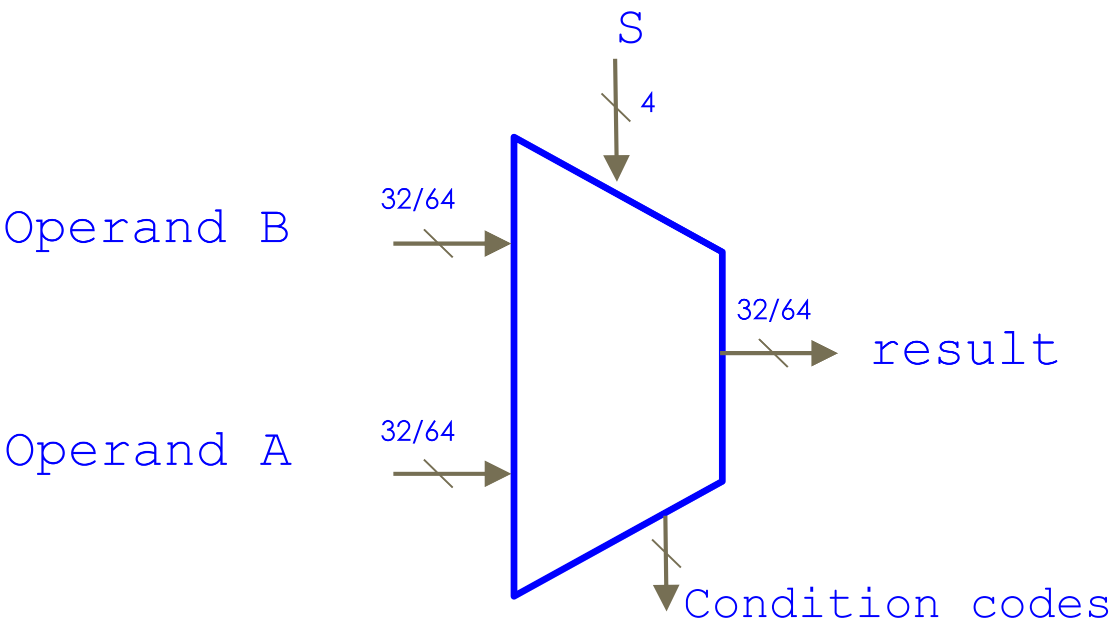


- Why is the propagation delay of this Full Adder 4 (also known as a 4-bit (w-bit) ripple carry adder) 9 gates (2w + 1 gates) where w is the number of bits (as opposed to 12 gates -> 4 x 3 p.d.)?

gate

ALU - Example of combinational logic circuit

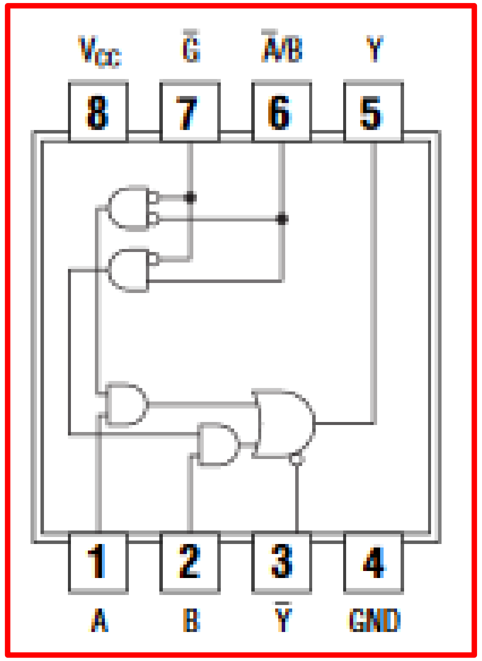
- Arithmetic/logic unit found in microprocessor
  - Black box diagram
- ALU executes *data manipulation machine instructions* (defined in the ISA)



S	**operation**
0001	**A + B**
0010	**A – B**
1000	**A * B**
0100	**A ^ B**
0101	**A & B**

Summary: Combinational logic circuits

- **Definition**: A combinational logic circuit computes a function, where its outputs is based only on its inputs
- Contains logic gates
  - Input values propagate through logic gates of combinational logic circuit whenever they change
  - Input to a logic gate must come from either input of combinational logic circuit itself (input to black box) or output of other logic gates
  - Outputs of two or more logic gates cannot be connected together unless it is via a logic gate
- Acyclic: there are no feedback loops
- Does not require a clock signal, does not contain memory elements -> does not store (remember) anything



From our last lecture - Digital circuits

Types of components found in a digital system such as a microprocessor

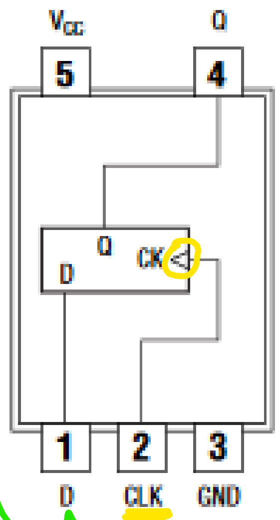
- In order to understand how the microprocessor executes these machine instructions (series of 0's and 1's), we need to have a look at the components of a microprocessor and how they function:
  1. Combinational logic -> manipulate bits (compute functions on bits e.g., ADD)
  2. Memory elements -> store bits
  3. Clock signals -> regulate the update of memory elements

  Sequential logic circuit

  and what affects the execution speed of these components such as propagation delay
- So, we need to understand a few things about ***digital circuits***

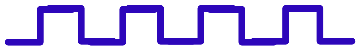
Sequential logic circuit

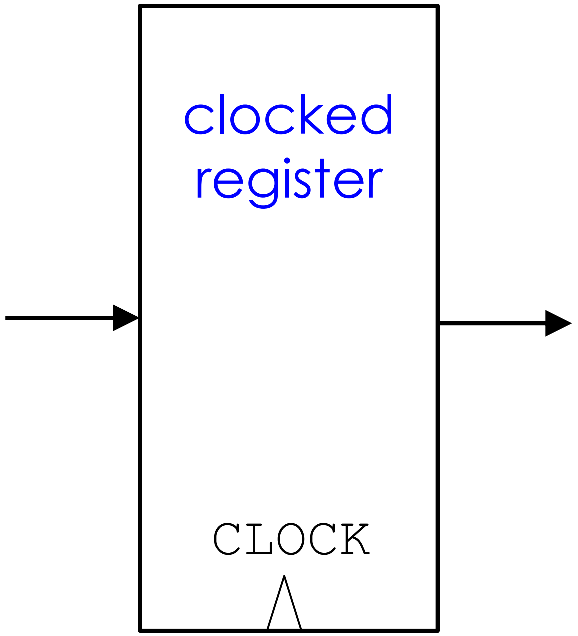
- Has a state -> memory (it has internal storage)
- "Clocked"
  - Has at least 2 input:
    - Data to be remembered and
    - Clock
  - Has at least one output:
    - Value remembered (during an earlier clock cycle)
- Example of a sequential logic circuit is a memory element
- Simplest memory element: D-type flip-flop
- Example of memory element components with which we are already familiar: registers

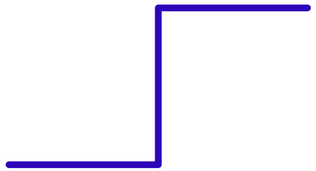


1G79
SINGLE POSITIVE-EDGE-TRIGGERED D-TYPE FLIP-FLOP

Memory elements and Clock signals

- Memory element #1 -> clocked registers a.k.a. hardware registers
  - On the microprocessor
  - A clocked register stores 1 bit (state)
  - Synchronized by system-wide clock
- System-wide clock 
  - A system-wide clock sends 0 1 0 1 0 1 0 1…
  - Clock period: 1 full cycle duration: 
  - Clock frequency: 1/period
- How clocked registers work:
  - Output current state
  - Input next state
  - Next state remembered only on rising edge of clock

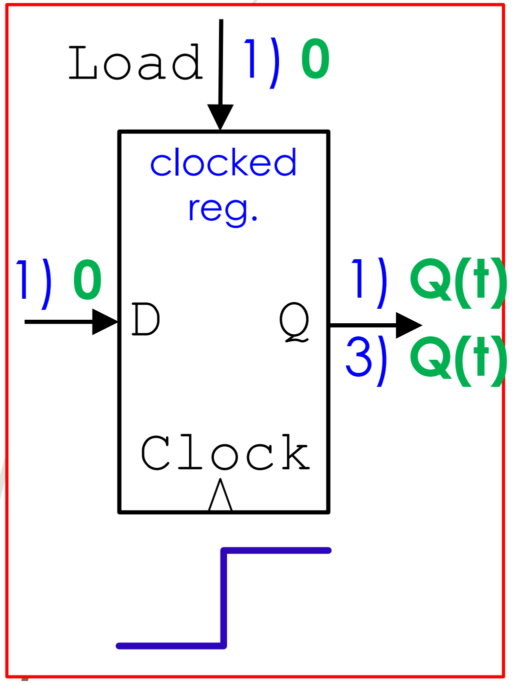




How a *clocked register* function!

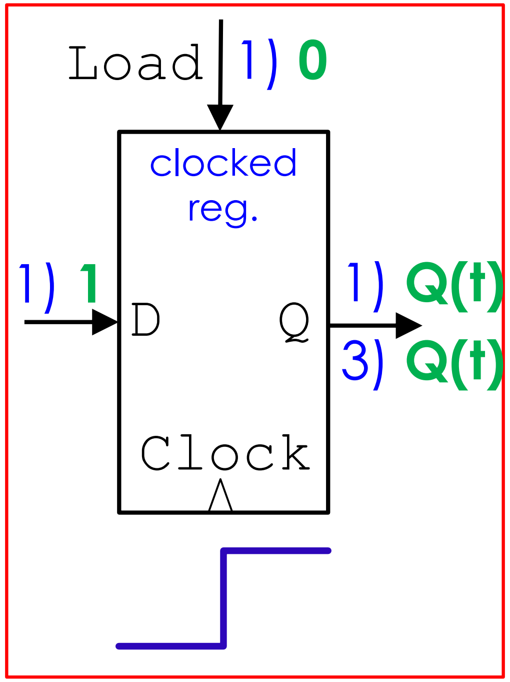
Function Table

Load	D	Q(t + 1) -> next state
0	X	Q(*t*) -> current state
1	0	0
1	1	1



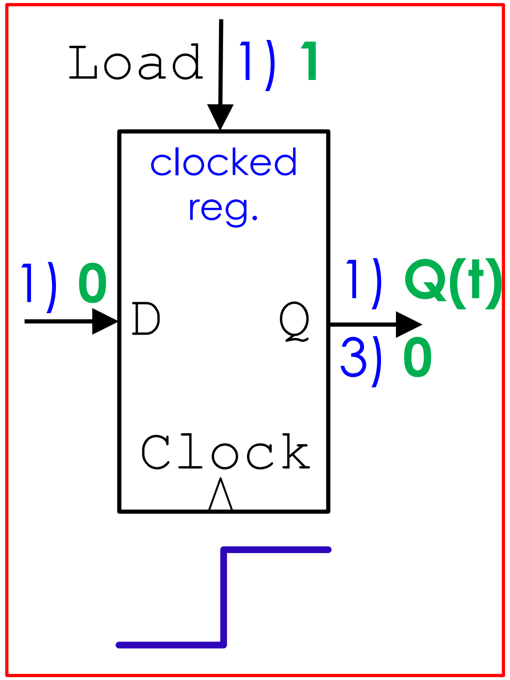
2) When the clock "ticks", i.e., when edge of clock rises

Row 1 of Function Table



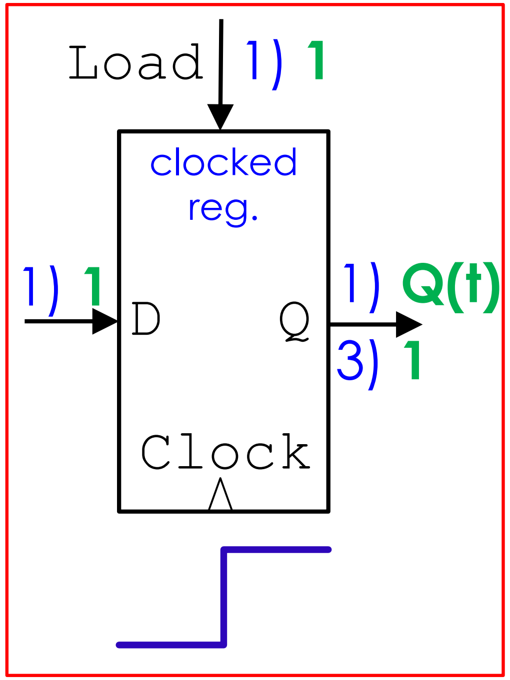
2) When the clock "ticks", i.e., when edge of clock rises

Row 1 of Function Table



2) When the clock "ticks", i.e., when edge of clock rises

Row 2 of Function Table



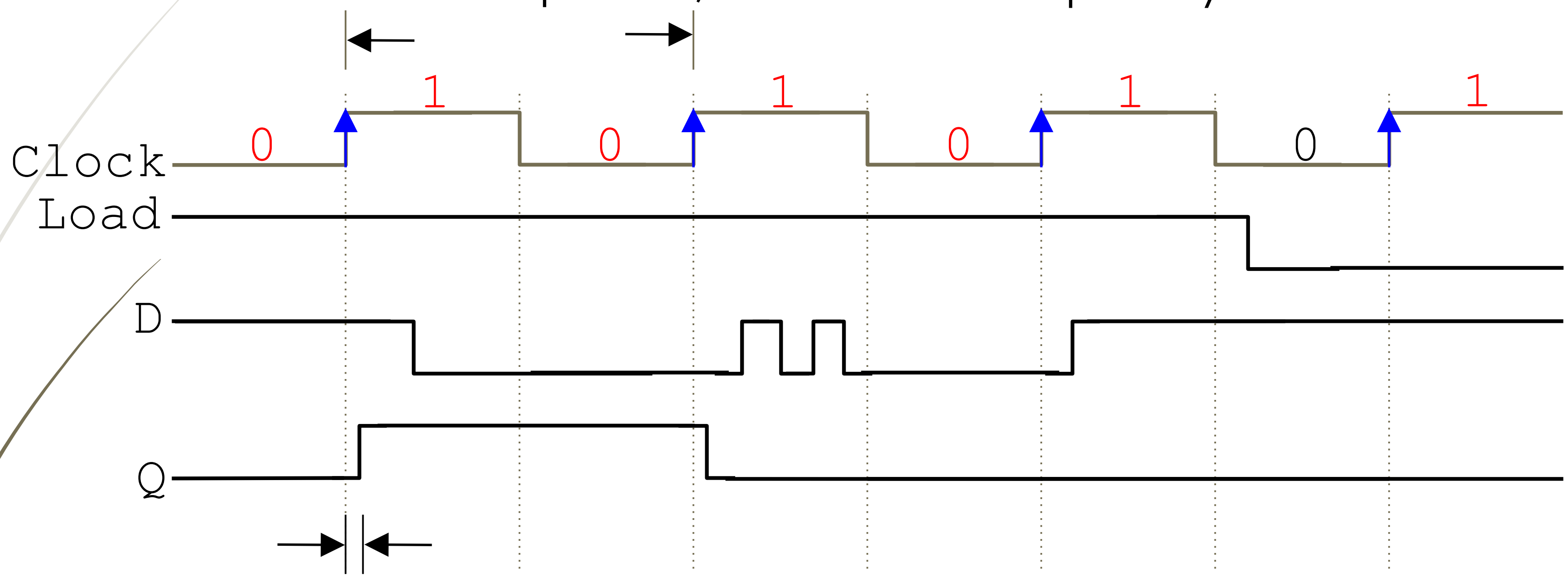
2) When the clock "ticks", i.e., when edge of clock rises

Row 3 of Function Table

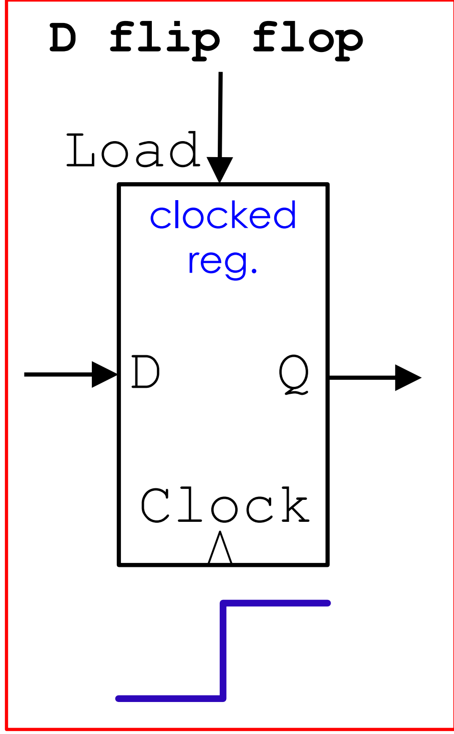
Timing Diagram

Edge-triggered clock

If clock period, then clock frequency is



t_{pd} for register



Function Table

Load	**D**	**Q(*t* + 1)**
0	X	Q(*t*)
1	0	0
1	1	1

Summary

- ✓ Combinational logic circuits
  - Made of many logic gates
  - Multi-functional combinational logic circuits such as ALU have control input lines to indicate which function to perform
  - Combinational logic circuits do not store (remember) values
- ✓ Sequential logic circuits
  - Made of combinational logic circuits, memory elements (clocked registers) and clock
  - Circuit that "remembers" values (state) and perform computations on these values

Next Lecture

- Instruction Set Architecture (ISA)
  - Definition of ISA
- Instruction Set design
  - Design principles
  - Look at an example of an instruction set: MIPS
  - Create our own
  - ISA evaluation
- Implementation of a microprocessor (CPU) based on an ISA
  - Execution of machine instructions (datapath)
  - Intro to logic design + Combinational logic + Sequential logic circuit
  - Sequential execution of machine instructions
  - Pipelined execution of machine instructions + Hazards