Binary Arithmetic I

CSCI 2050U - Computer Architecture

Randy J. Fortier
@randy_fortier

Outline

- Binary addition
- Half adder
- Full adder
- Ripple carry adder
- Fast carry adder

Binary Addition

CSCI 2050U - Computer Architecture

Adding Binary Numbers

- Adding two binary numbers is very similar to adding two decimal numbers
  - Except that the digits cannot exceed 1 (unlike decimal, where 9 is the limit)

```
  1001 0101
+ 0001 1100
  ---------
```

Adding Binary Numbers

- Adding two binary numbers is very similar to adding two decimal numbers
  - Except that the digits cannot exceed 1 (unlike decimal, where 9 is the limit)

```
  1001 0101          (1+0=1)
+ 0001 1100
  ---------
          1
```

Adding Binary Numbers

- Adding two binary numbers is very similar to adding two decimal numbers
  - Except that the digits cannot exceed 1 (unlike decimal, where 9 is the limit)

```
  1001 0101        (0+0=0)
+ 0001 1100
  ---------
         01
```

Adding Binary Numbers

- Adding two binary numbers is very similar to adding two decimal numbers
  - Except that the digits cannot exceed 1 (unlike decimal, where 9 is the limit)

```
        1
  1001 0101        (1+1=2, in binary 2 is 10)
+ 0001 1100
  ---------
         001
```

Adding Binary Numbers

- Adding two binary numbers is very similar to adding two decimal numbers
  - Except that the digits cannot exceed 1 (unlike decimal, where 9 is the limit)

```
     1 1
  1001 0101
+ 0001 1100
  ---------
       0001
```

Adding Binary Numbers

- Adding two binary numbers is very similar to adding two decimal numbers
    - Except that the digits cannot exceed 1 (unlike decimal, where 9 is the limit)

```
     11 1
   1001 0101          (1+1+1=3, in binary 3 is 11)
 + 0001 1100
   ---------
      1 0001
```

Adding Binary Numbers

- Adding two binary numbers is very similar to adding two decimal numbers
  - Except that the digits cannot exceed 1 (unlike decimal, where 9 is the limit)

```
     11 1
   1001 0101
 + 0001 1100
   ---------
     11 0001
```

Adding Binary Numbers

- Adding two binary numbers is very similar to adding two decimal numbers
  - Except that the digits cannot exceed 1 (unlike decimal, where 9 is the limit)

```
    11  1
  1001 0101
+ 0001 1100
  ---------
   011 0001
```

Adding Binary Numbers

- Adding two binary numbers is very similar to adding two decimal numbers
  - Except that the digits cannot exceed 1 (unlike decimal, where 9 is the limit)

```
    11  1
  1001 0101
+ 0001 1100
  ---------
  1011 0001
```

Adding Binary Numbers

- Adding two binary numbers is very similar to adding two decimal numbers
  - Except that the digits cannot exceed 1 (unlike decimal, where 9 is the limit)

```
     11 1
  1001 0101     149
+ 0001 1100     +28
  ---------     ---
  1011 0001     177
```

Adding Binary Numbers

- If there is a carry, because computers store data in finite spaces, we can't add any extra digits
  - There is no way to correctly encode the result
  - We have to throw away the carry, and the result will be wrong
  - This is called overflow

```
 1   1   1 1
  1001 0101    149
+ 1001 0101    149
  ---------    ---
  0010 1010     42 (incorrect)
```

Adder Circuits

CSCI 2050U - Computer Architecture

Ontario Tech University

Half Adder

- A half adder (HA) can add two bits together
  - Inputs:
    - `A` and `B`: The two bits to add
  - Outputs:
    - `S`: The sum of the two bits
    - `C`: Carry (1 if there was a carry, 0 if not)

Half Adder

- A half adder can add two bits together

A	B	S	C
0	0		
0	1		
1	0		
1	1		

Half Adder

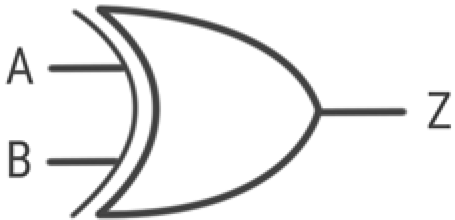
- A half adder can add two bits together
  - This truth table is that of the XOR gate: $S = A'B + AB' = A \oplus B$

A	B	S	C
0	0	0	
0	1	1	
1	0	1	
1	1	0	

Exclusive OR

- XOR:

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



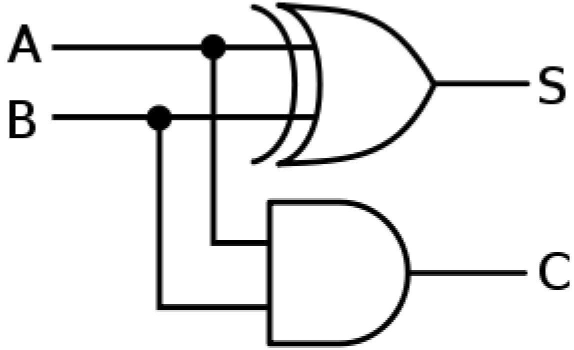
Half Adder

- A half adder can add two bits together
  - Recognize this truth table? `C = AB`

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

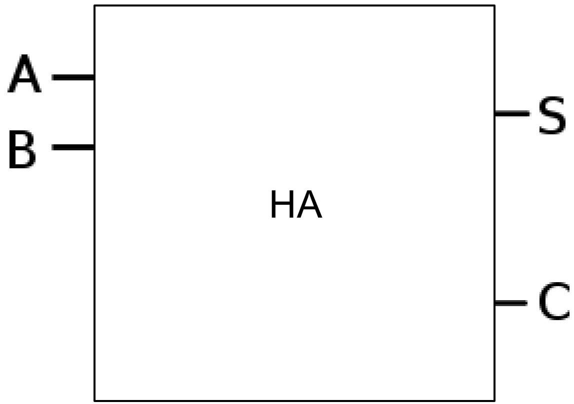
Half Adder

- A half adder circuit:
  - $S = A'B + AB' = A \oplus B$
  - $C = AB$



Half Adder

- A half adder in block notation:



Full Adder

- Why do they call it a half adder?
- To find the answer to this, we need to try to add two multi-bit numbers
- Let’s add `0011` and `0111`:

```
 0011
+0111
-----
```

Full Adder

- Why do they call it a half adder?
- To find the answer to this, we need to try to add two multi-bit numbers
- Let's add `0011` and `0111`:
    - Adding the rightmost two bits can be done with a half adder

```
   1
 0011
+0111
-----
    0
```

Full Adder

- Why do they call it a half adder?
- To find the answer to this, we need to try to add two multi-bit numbers
- Let's add `0011` and `0111`:
  - Adding the rightmost two bits can be done with a half adder
  - What about the next two bits?

```
  1 1
 0011
+0111
-----
   10
```

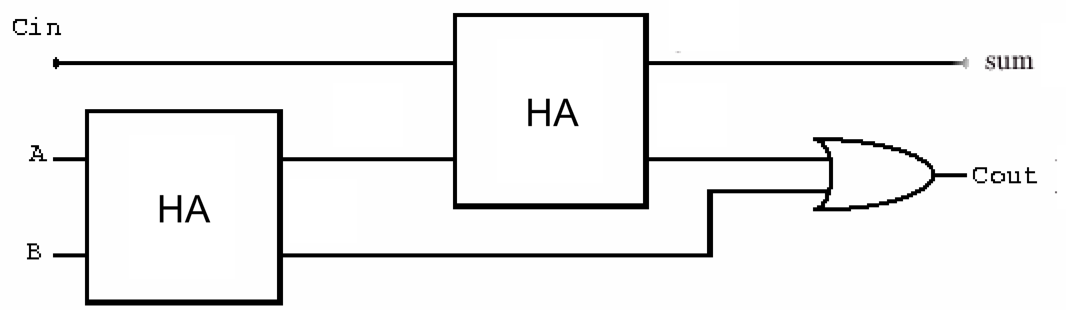
Full Adder

- Why do they call it a half adder?
- To find the answer to this, we need to try to add two multi-bit numbers
- Let's add `0011` and `0111`:
  - Adding the rightmost two bits can be done with a half adder
  - What about the next two bits?
    - We really need to add three bits, in the general case
    - For this we need a new component: the full adder (FA)

```
  1 1
 0011
+0111
-----
   10
```

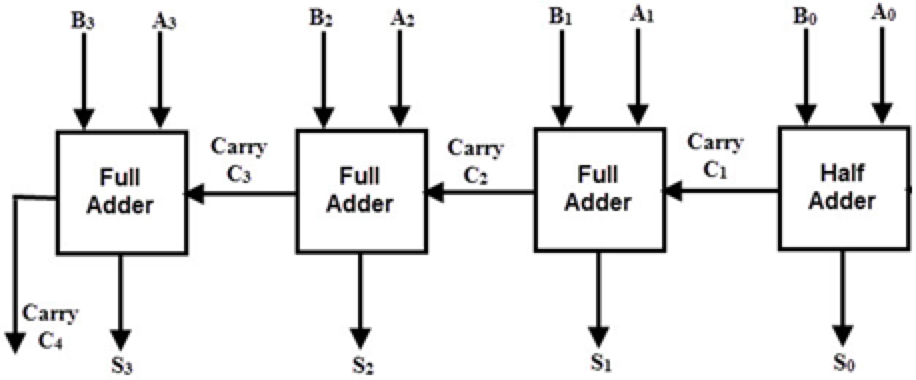
Full Adder

- A full adder (FA) adds three bits
  - Inputs:
    - A and B: The bits to be added
    - C_{in}: A carry from a previous add, which also must be added
  - Outputs:
    - S: The sum of the three bits
    - C_{out}: Whether or not the sum resulted in a carry bit



Ripple Carry Adder

- We could have drawn this entire circuit using XOR, AND, and OR gates
- This is our first circuit, drawn using block notation
- This is called a *ripple carry adder*



Fast Carry Adder

- A ripple carry adder requires that the carry propagates from one adder to the next
  - For a 64-bit adder, this would take 64 clock cycles
- A fast carry adder is exactly the same, but calculates the carry in inputs separately so that each full adder can determine its sum in the first clock cycle

Wrap-up

- Binary arithmetic
- Half adder
- Full adder
- Ripple carry adder
- Fast carry adder

What is next?

- Signed number representations
- Binary subtraction
- Overflow