Mux, Binary Arithmetic

CS 2130: Computer Systems and Organization 1

Xinyao Yi Ph.D.
Assistant Professor

UNIVERSITY of VIRGINIA | ENGINEERING

Multiplexer (mux)

x=a ? b : c

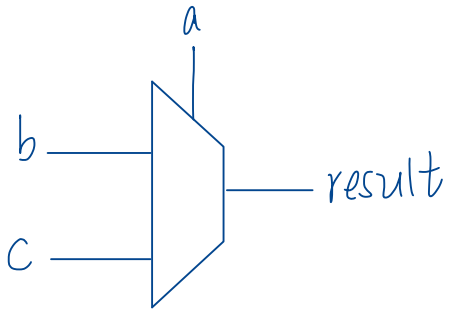
A multiplexer (mux) is commonly drawn as a trapezoid in circuit diagrams.

Multiplexer (mux)

ternary operator

x=a ? b : c

A multiplexer (mux) is commonly drawn as a trapezoid in circuit diagrams.




```
if(a){
    b
}else{
    c
}
```

a	b	c	result
0			c (depends on c)
1			b (depends on b)

a, b, c could be anything, they could be strings, numbers, functions.....

a.b.c are all one bit.

a	b	c	result
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

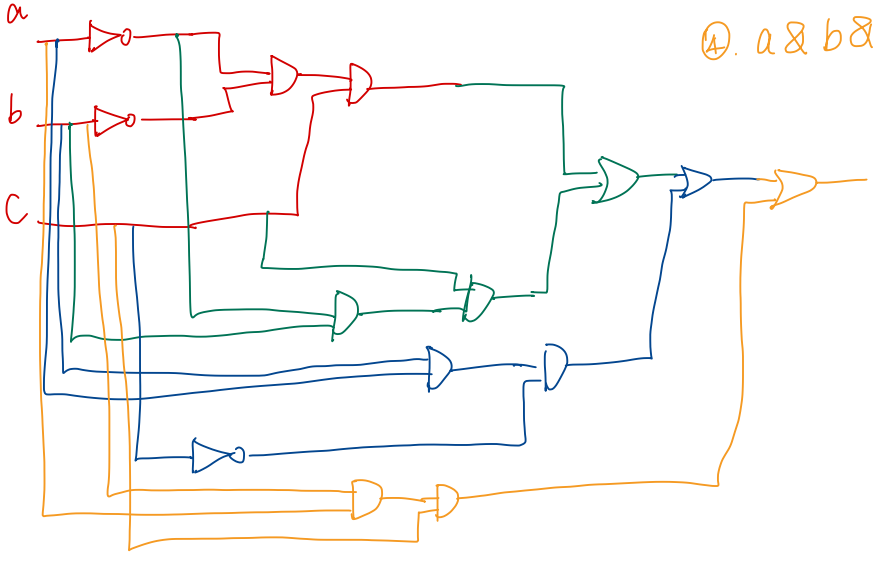
when the result is true ?

(!a&!b&c) | (!a&b&c) |

(a&b&!c) | (a&b&c)

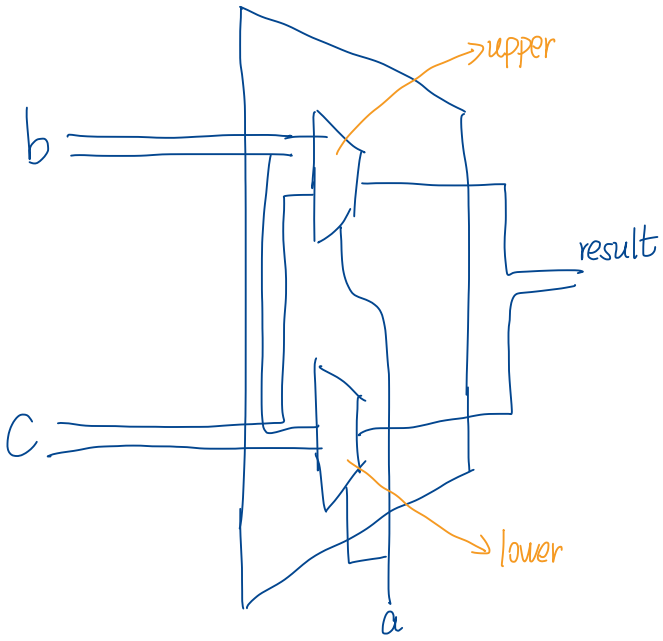
① !a&!b&c. ②. !a&b&c. ③. a&b&!c.

④. a&b&c



2-bit Multiplexer (mux)

2-bit values instead of 1-bit values



Multi-bit Values

So far, only talking about 2 things: 0 and 1

Next:

Numbers, strings, objects, ...

Numbers

From our oldest cultures, how do we mark numbers? ||||| For small numbers: ok. But for big numbers? X

- unary representation: make marks, one per "thing"
  - Awkward for large numbers, ex: CS 2130?
  - Hard to tell how many marks there are
- Update: group them! 𝍸 𝍸 maybe easier to count, but still has issues when present big numbers.
- Romans used new symbols:

5	50	100	500	1000
V	L	C	D	M

(at least get this shorter).

Numbers

Arabic numerals

- Positional numbering system

the position means something.

decimals:

2000	100	30	0
↑	↑	↑	↑
2	1	3	0
↑	↑	↑	↑
10^3	10^2	10^1	10^0

$2\times1000 + 1\times100 + 3\times10 + 0\times1 = 2130$

Numbers

Arabic numerals

- Positional numbering system
- The 10 is significant:
  - 10 symbols, using 10 as base of exponent

Numbers

Arabic numerals

- Positional numbering system
- The 10 is significant:
  - 10 symbols, using 10 as base of exponent
- The 10 is arbitrary
  - We can use other bases! π, 2130, 2, ... → binary

Base-8 Example

Try to turn 134_8 into base-10:

		1	3	4
8^4	8^3	8^2	8^1	8^0
		64	8	1

$1 \times 64 + 3 \times 8 + 4 \times 1 = 92_{10}$

Bases

We will discuss a few in this class

- Base-10 (decimal) - talking to humans
- Base-8 (octal) - shows up occasionally
- Base-2 (binary) - most important! (we've been discussing 2 things!)
- Base-16 (hexadecimal) - nice grouping of bits

Binary

2 digits: 0, 1

Try to turn 1100101_2 into base-10

1	1	0	0	1	0	1
2^6	2^5	2^4	2^3	2^2	2^1	2^0
64	32	16	8	4	2	1

$64 + 32 + 4 + 1 = 101_{10}$ (only additions needed).

Binary

Any downsides to binary?

Turn 2130_{10} into base-2:

hint: find largest power of 2 and subtract

2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
2048	1024	512	256	128	64	32	16	8	4	2	1
1	0	0	0	0	1	0	1	0	0	1	0

2130 - 2048 = 82 82 - 64 = 18 18 - 16 = 2

Binary

Any downsides to binary?

Turn 2130_{10} into base-2:

hint: find largest power of 2 and subtract

2 | 2130
2 | 1065 — 0
2 | 532 — 1
2 | 266 — 0
2 | 133 — 0
2 | 66 — 1
2 | 33 — 0
2 | 16 — 1
2 | 8 — 0
2 | 4 — 0
2 | 2 — 0
2 | 1 — 0
0 — 1

100001010

Long Numbers

How do we deal with numbers too long to read?

Long Numbers

How do we deal with numbers too long to read?

- Group them by 3 (right to left)

Long Numbers

How do we deal with numbers too long to read?

- Group them by 3 (right to left)
- In decimal, use commas: ,
- Numbers between commas: 000 – 999
- Effectively base-1000 → You can think about it from this way:

1,234,567

↑ ↑ ↑

1000^2 1000^1 1000^0

$567 \times 1000^0 + 234 \times 1000^1 + 1 \times 1000^2$

Long Numbers in Binary - Readability

- Typical to group by 3 or 4 bits
- No need for commas *Why?* In decimal, we use commas every 3 digits, because place values repeat every power of 1000. in binary, grouping by 3 or 4 already gives us a fix-size block that maps to another digit system.

100001010010

Long Numbers in Binary - Readability

- Typical to group by 3 or 4 bits
- No need for commas *Why?*
- We can use a separate symbol per group
- How many do we need for groups of 3?

How many numbers can be present using 3 bits?

000 – 111

0, 1, 2, 3, 4, 5, 6, 7 (8) $2^3 = 8$

100001010010

Long Numbers in Binary - Readability

- Typical to group by 3 or 4 bits
- No need for commas *Why?*
- We can use a separate symbol per group
- How many do we need for groups of 3?
- Turn each group into decimal representation

100001010010

Long Numbers in Binary - Readability

- Typical to group by 3 or 4 bits
- No need for commas *Why?*
- We can use a separate symbol per group
- How many do we need for groups of 3?
- Turn each group into decimal representation
- Converts binary to **octal**

100001010010

4 1 2 2 8

Long Numbers in Binary - Readability

- Groups of 4 more common
- How many symbols do we need for groups of 4?

$2^4 = 16$

100001010010

8 5 2 $_{16}$

Long Numbers in Binary - Readability

- Groups of 4 more common
- How many symbols do we need for groups of 4?
- Converts binary to **hexadecimal**
- Base-16 is very common in computing

100001010010

Hexadecimal

Need more than 10 digits. What next?

1110

$2^3 2^2 2^1 2^0$

8 4 2 1

8+4+2 = 14

↓

E

1
2
3
4
5
6
7
8
9
a → 10
b → 11
c → 12
d → 13
e → 14
f → 15

Hexadecimal Exercise

Consider the following hexadecimal number:

852dab1e

Is it even or odd?

See the last digit.