Review

ECE 3710

No one knows what's next,
but everybody does it.

- George Carlin

for what follows...

Most Likely You Go Your Way (And I'll Go Mine)

- Bob Dylan

binary

2^n	n
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16

8 bits: 0000 $\underbrace{0000}$

2^8 = 256 => 256 representations

bin2dec

2^n	n
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16

0

1111 1111

7

2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0 = 255

bin2dec

Q: 0101 0101

bin2dec

Q: 0101 0101

A: 2^6+2^4+2^2+2^0 = 85

never, ever use this:

"There are `10` types of people: those who understand binary and everyone else."

if you think it’s funny...

dec2bin

275 => 1 0001 0011

bits:

1. how many
2. which

dec2bin

2^n	n
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16

275 => 1 0001 0011

275 > 255 => more than 8

275 < 511 => less than 9

dec2bin

2^n	n
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16

275 => 1 0001 0011

tedious

$$2^8 + \underbrace{275-256}$$

19 = 1 0011

fastest

2^8 + 2^(i-1) <= 275

hex

Dec	Hex	Bin
00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

hex2dec & dec2hex

use bin as intermediary

2^n	n
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16

Q: dec2bin(34) & dec2hex(34)

Dec	Hex	Bin
00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Q: dec2bin(34) & dec2hex(34)

2^5 = 32 and 2^6=64 => 6 bits: 0010 0000

2^n	n
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15

32 + 2 = 34 => 0010 0010

22H

2^n	n
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16

Q: hex2dec(A5)

Dec	Hex	Bin
00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Q: hex2dec(A5)

$\downarrow$

10 5

$\downarrow$

shouldn't need table: 1010 0101

(memorise all 4-bit)

$\downarrow$

$2^7+2^5+2^2+2^0=128+32+4+1=165$

2^n	n
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16

Q: 36BH+F6H

Dec	Hex	Bin
00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Q: 36BH+F6H

11
36B
\+ F6
=461

1: 11+6=17 (-16=1)
2:6+15+1=22 (-16=6)
3: 3+1=4

2^n	n
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16

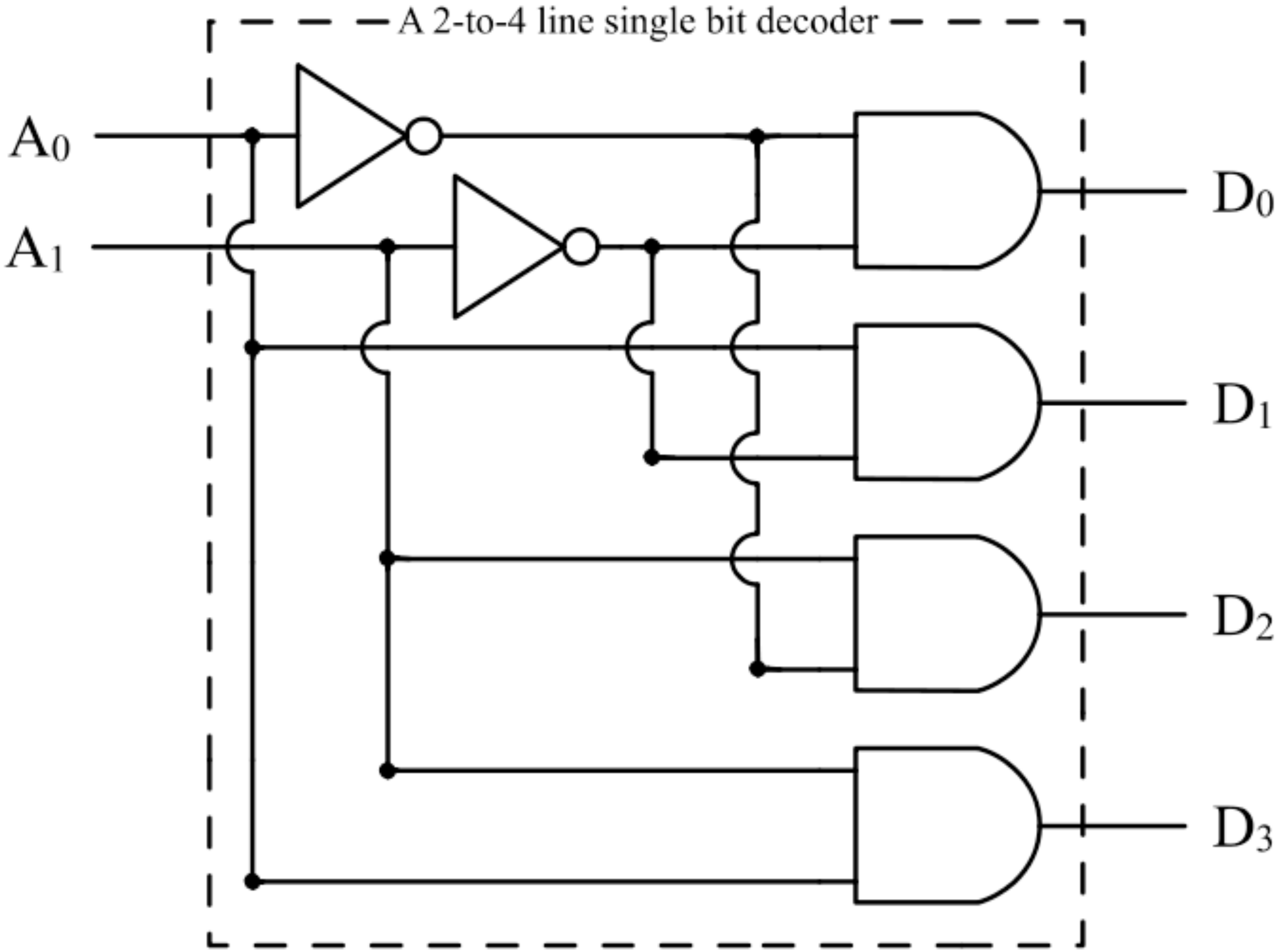
Q: 36BH-F6H

Dec	Hex	Bin
00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Q: 36BH-F6H

36B - F6 =275	1: 11-6=5 2:6-15+16=7 3: 2

decoder



Truth Table

A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Minterm Equations

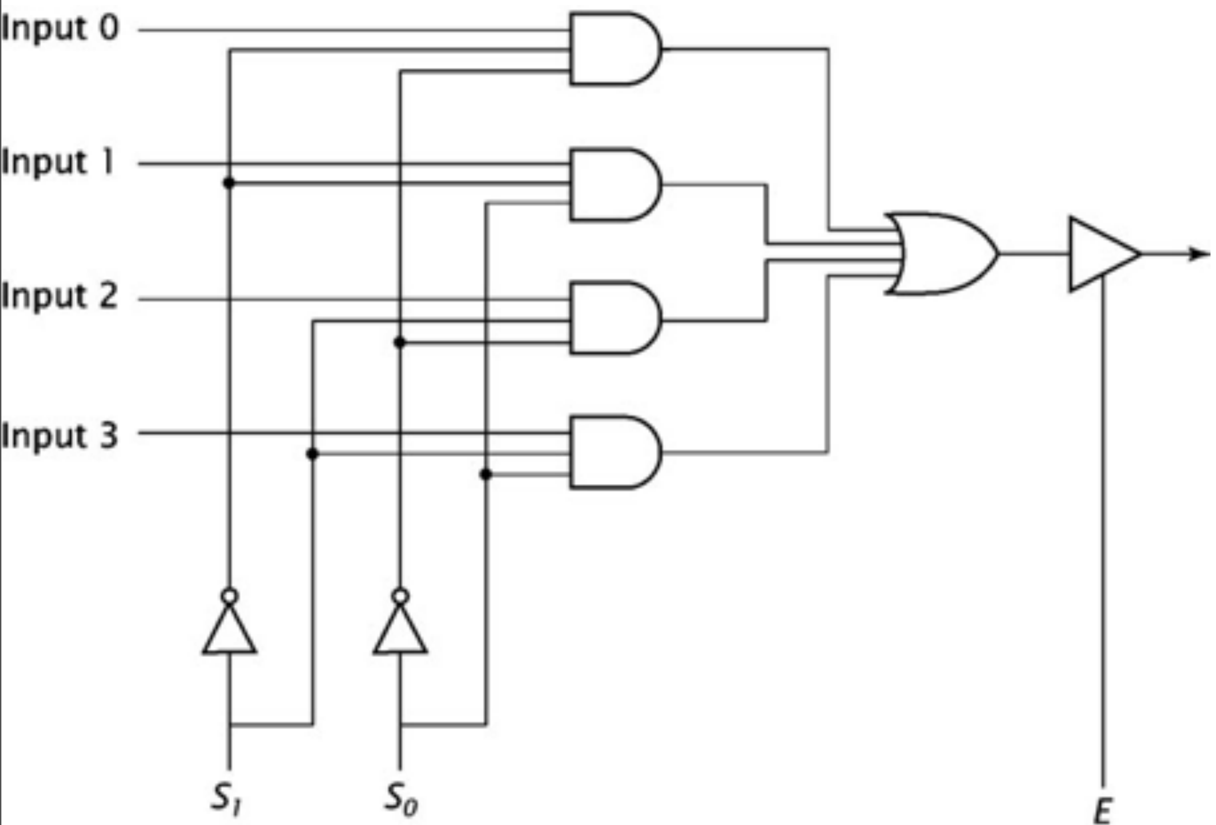
$$D_0 = \overline{A_1} \cdot \overline{A_0}$$

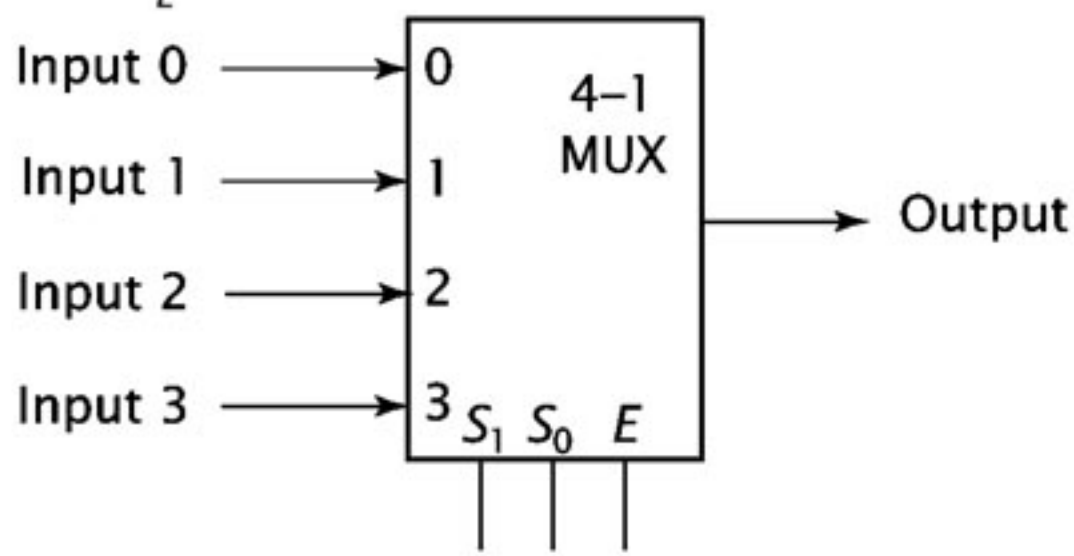
$$D_1 = \overline{A_1} \cdot A_0$$

$$D_2 = A_1 \cdot \overline{A_0}$$

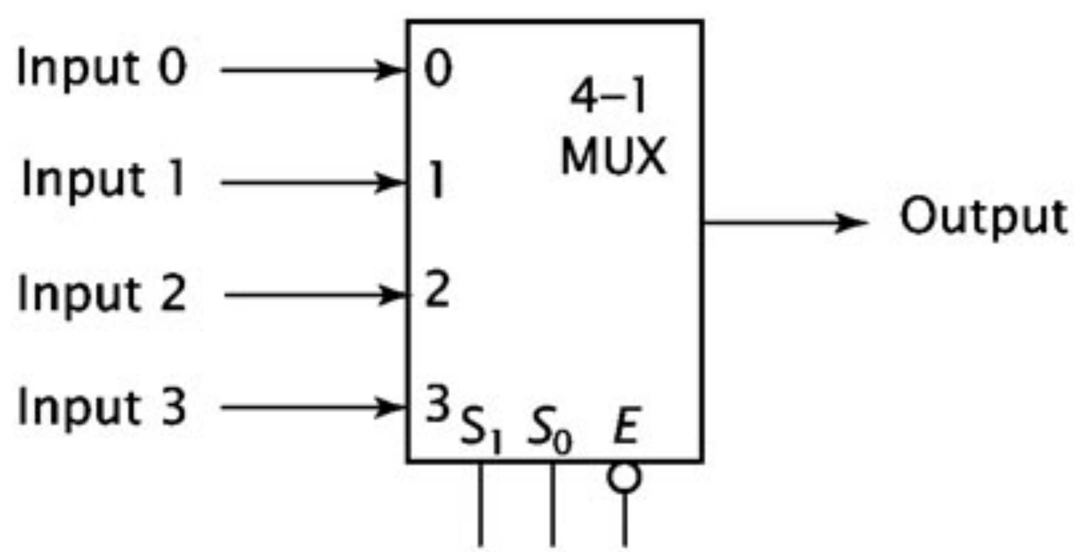
$$D_3 = A_1 \cdot A_0$$

multiplexor





S_1	S_0	E	Output
X	X	0	Z
0	0	1	Input 0
0	1	1	Input 1
1	0	1	Input 2
1	1	1	Input 3



S_1	S_0	E	Output
X	X	1	Z
0	0	0	Input 0
0	1	0	Input 1
1	0	0	Input 2
1	1	0	Input 3

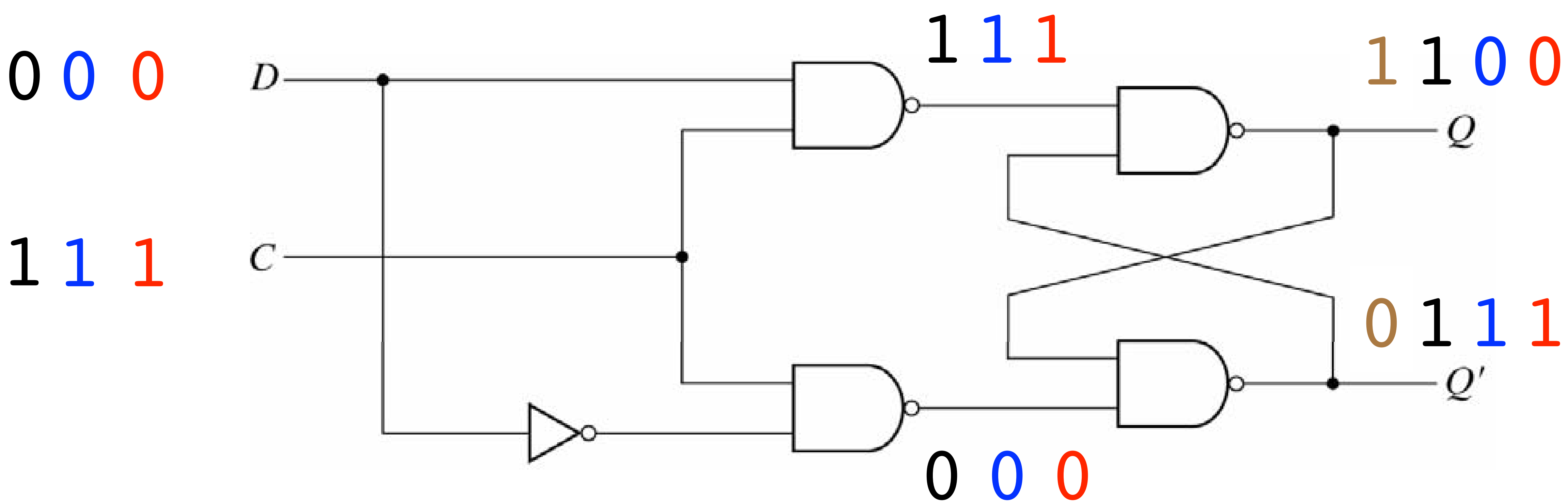
flip-flop (d latch)

C	D	Next state of Q
0	X	No change
1	0	$Q = 0$; Reset state
1	1	$Q = 1$; Set state

previous clock: $Q = 1$

$Q' = 0$

want: $Q = 0$



who makes a 112 GB HDD?

Default

```
m1:~ rgerdes$ df -h
Filesystem      Size   Used  Avail Capacity  Mounted on
/dev/disk1     112Gi  106Gi  6.2Gi    94%    /
devfs          187Ki  187Ki    0Bi   100%    /dev
map -hosts       0Bi    0Bi    0Bi   100%    /net
map auto_home    0Bi    0Bi    0Bi   100%    /home
m1:~ rgerdes$ df -H
Filesystem      Size   Used  Avail Capacity  Mounted on
/dev/disk1      120G   113G   6.6G    94%    /
devfs           191k   191k     0B   100%    /dev
map -hosts        0B     0B     0B   100%    /net
map auto_home     0B     0B     0B   100%    /home
m1:~ rgerdes$
```

I bought a 120 GB HDD

Why is 1 GB equal to 10^9 bytes instead of 2^30?

Because in 1960, the Bureau International des Poids et Mesures decided that the SI prefix G- meant 10^9.

But it means 2^30, really!

No it doesn't. Let's look at some examples:

A Gm is...	10^9 meters
A GW is...	10^9 watts
A GA is...	10^9 amperes
A Gmol is...	10^9 moles

But it's different in computing!

Let's look at some more examples:

A 2.2 GHz CPU operates at...	2.2 x 10^9 cycles per second
1 Gbps Ethernet transmits data at...	10^9 bits per second
The 2.4GHz band which wireless ethernet operates within lies...	between 2.4 x 10^9 and 2.5 x 10^9 Hz
A 200 GB hard drive holds...	200 x 10^9 bytes of data

But what about RAM?

You're right: If you buy a "1GB" stick of RAM, it will hold 2^30 bytes of data.

However, this is a special case: Unlike everything else in the world of computing, RAM is addressed in hardware. When you're designing a piece of silicon, you want to have N address lines and have every combination of zeroes and ones map to a memory location — to do otherwise would make the logic far more complicated. *Nothing else* is addressed this way.

Finally, even for RAM calling 2^30 bytes "1GB" isn't really proper; instead, the IEC binary multiplier prefix "Gi-" should be used.

This is all a conspiracy by hard drive manufacturers who want to cheat us out of the disk space we're paying for!

We all love good conspiracy theories... but really, this isn't about evil megacorporations trying to cheat you. Hard drive prices are determined almost entirely by competition between manufacturers, so if hard drives were labelled in GiB instead of being labelled in GB, we'd be paying the same number of dollars for the same number of bytes anyway — if this really was a global conspiracy, it would be one of the dumbest conspiracies ever.

http://www.tarsnap.com/GB-why.html