University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 120: Introduction to Computing

Extra Slides for Spring 2017

I Need Your Help

Do you ever watch movies?

My friends want to have a movie club.

There are three coming out that we're considering…

2/10 LEGO Batman … animation, funny
3/10 Kong: Skull Island … thriller
4/7 Going in Style … bank heist

The problem: we're all pretty picky. If we can't agree in advance which ones to watch, we won't do the club.

Which Movies Should We Watch?

2/10 LEGO Batman … animation, funny
3/10 Kong: Skull Island … thriller
4/7 Going in Style … bank heist

Me: Kong is out earlier in Europe, so I will catch it there. I do want to watch at least one movie.

Jan: Three is too many.

Alice: Let's watch exactly one that isn't an animation.

Bob: I love Batman, so we have to see that one.

Can You Translate My Needs To Boolean?

L = 2/10 LEGO Batman … animation, funny
K = 3/10 Kong: Skull Island … thriller
G = 4/7 Going in Style … bank heist

Me: Kong is out earlier in Europe, so I will catch it there. I do want to watch at least one movie.

$$K' \; (L + K + G)$$

Can You Translate Jan's Needs To Boolean?

L = 2/10 LEGO Batman … animation, funny
K = 3/10 Kong: Skull Island … thriller
G = 4/7 Going in Style … bank heist

Jan: Three is too many.

(L' + K' + G')

Can You Translate Alice's Needs To Boolean?

L = 2/10 LEGO Batman … animation, funny
K = 3/10 Kong: Skull Island … thriller
G = 4/7 Going in Style … bank heist

Alice: Let's watch exactly one that isn't an animation.

(K⊕G)

Can You Translate Bob's Needs To Boolean?

L = 2/10 LEGO Batman … animation, funny
K = 3/10 Kong: Skull Island … thriller
G = 4/7 Going in Style … bank heist

Bob: I love Batman, so we have to see that one.

L

Which Movies Should We Watch?

Me: K'(L+K+G)
Jan: (L'+K'+G')
Alice: (K⊕G)
Bob: L

We need to satisfy ALL four people… AND!

So: Lego and Going in Style…thanks!

L	K	G	Movies
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Long Definition for Overflow of 2's Complement Addition

Recall the overflow condition **V** for **2's complement** addition.

Add two **N-bit 2's complement** patterns.

$$\begin{array}{rl} & \mathbf{A}\ a_{N-2} \ldots a_0 \text{ (sign bit is A)} \\ + & \mathbf{B}\ b_{N-2} \ldots b_0 \text{ (sign bit is B)} \\ \hline & \mathbf{S}\ s_{N-2} \ldots s_0 \text{ (sign bit is S)} \end{array}$$

We can calculate

$$V = ABS' + A'B'S$$

Another Way to Define 2's Complement Overflow?

Other lectures saw a different condition.
Let's first name two of the carry bits.

$$\begin{array}{rl} & C_N\ C_{N-1} \\ & \mathbf{A}\ a_{N-2} \ldots a_0 \text{ (sign bit is A)} \\ + & \mathbf{B}\ b_{N-2} \ldots b_0 \text{ (sign bit is B)} \\ \hline & \mathbf{S}\ s_{N-2} \ldots s_0 \text{ (sign bit is S)} \end{array}$$

The other lectures were then told that

$$V = C_N \oplus C_{N-1}$$

Are these two expressions the same?

One Proof Strategy: Algebra

We can use Boolean algebra to prove that

$V = ABS' + A'B'S$ equals $C_N \oplus C_{N-1}$

But it's not really so fun.

Trust me, I did it.

What about brute force? (a truth table)

We can calculate **S** and C_N from **A**, **B**, and C_{N-1}, so we only have 3 variables as "inputs."

Proof by Exhaustion / Brute Force

A	B	C_{N-1}	C_N	S	V	$C_N \oplus C_{N-1}$
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	1	0	0
0	1	1	1	0	0	0
1	0	0	0	1	0	0
1	0	1	1	0	0	0
1	1	0	1	0	1	1
1	1	1	1	1	0	0

Always Choose the Right Proof Strategy

Always choose the **clearest and fastest proof strategy** (usually those two metrics correlate).

Using brute force for proofs doesn't make you a brute!