Homework 5

Devlin Ih

2022-10-26

Context Beyond CompArch—Psychology/Information Theory

a) Describe at least one situation in which you had to use or measure more bits than were useful to a problem.

I have on many occasions been asked to rate things on a 0 to 10 scale, or sometimes 0 to 100 scale. 1 to 7 is realistically all that the typical person can give.

Here's another: Lab 1, Conway's Game of Life. The module `count_neighbors` takes in 8 bits of information and outputs 4 bits. We are counting between 0 and 8 (9 discrete values). This gives $\log_2 9 = 3.17$ bits of information. However, the ceiling log2 comes in, where 4 bits of information were used. In the end, 3 would actually be sufficient because the 4th bit became a don't care condition.

b) Have you ever had a situation where you didn't use enough bits?

Yes.

c) How many bits did you use to answer the previous question?

This is a similar question to when the paper discusses chunks. It could be 1 bit (yes/no). You could also say I used 3 letters, with each letter having a value of $\log_2 26 = 4.7$ bits; this is 14.1 bits. What about if you count capital and lowercase letters, that doubles it. Or this file has an 8-bit encoding; if you count the period in `Yes.` that gives 32 bits. Okay enough of this.

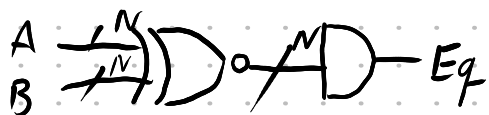d) In your own words, what is the difference between a bit and a chunk?

A bit is a quantity of something. A chunk is how we group bits of information into more meaningful values in memory for us. For example, our words are made of letters (26 of them, so each letter is 4.7 bits). But we can group these into chunks called words that store more information while being easier for us to remember.

e) How are the generalizations from this paper still applicable half a century later? Alternatively, what do you feel no longer applies?

Well, this is a rather shallow example, but using Morse Code -> letters -> words (types of chunks) seems a bit dated.

I think what is still definitely useful though is how 7 is the typical amount of discrete points we can distinguish. It makes sense to keep using 7 point scales and well, not ask silly 0 to 100 questions.

Compare Eq:

A [diagram: A and B inputs, buses labeled N, M, through XNOR/inverter and AND gate producing Eq]

$$Eq = \&\left(\sim(A \wedge B)\right)$$

Compare Lt:

overflow detection for subtraction

$\&$ — opposite signs for A and B $\quad (A[N-1] \wedge B[N-1])$
— opposite signs for sum and A $\quad (A[N-1] \wedge Sum[N-1])$

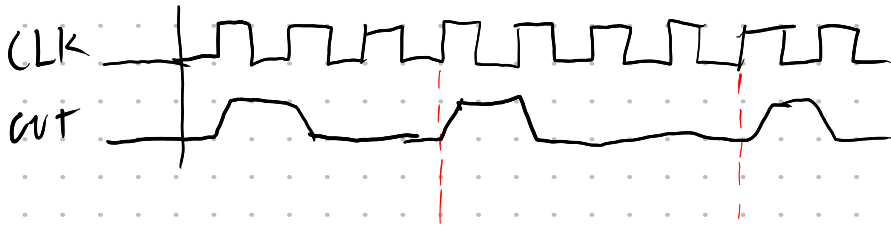$$V = (A[N-1] \wedge B[N-1]) \ \& \ (A[N-1] \wedge Sum[N-1])$$

According to H&H, for 2 signed numbers

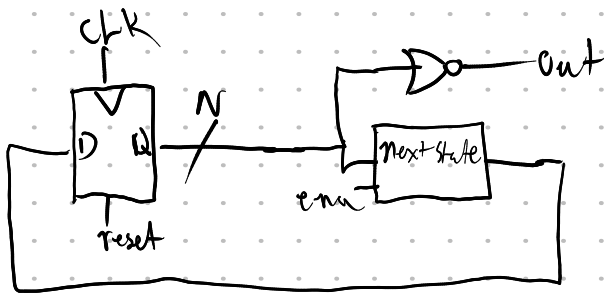$$L = V \wedge N \qquad N = Sum[N-1], \text{ negative output}$$
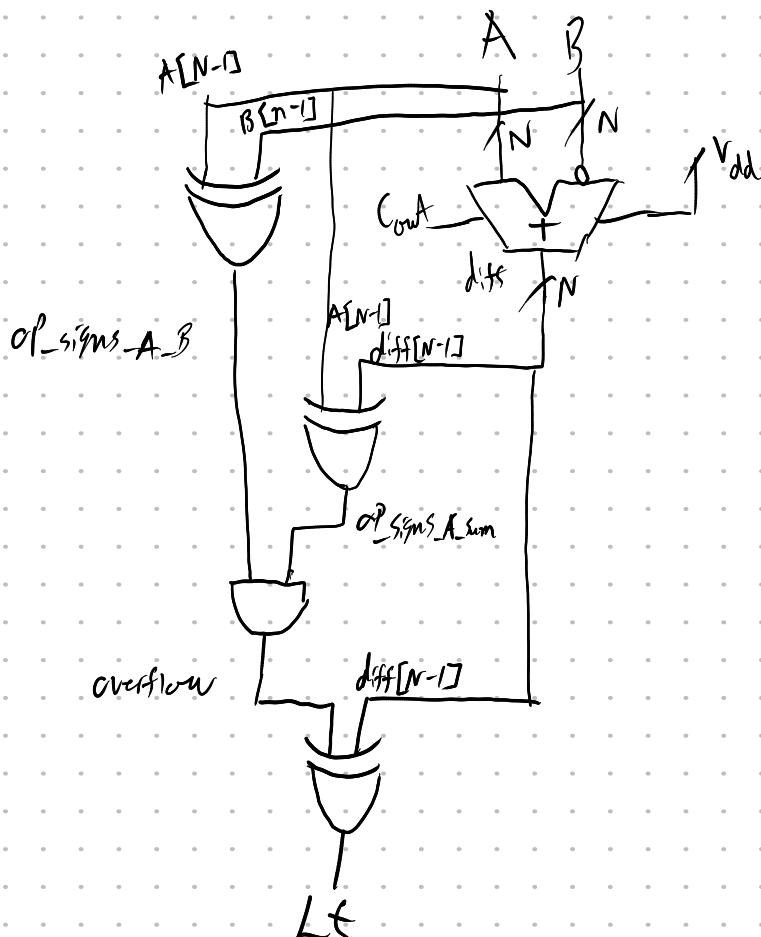
Pulse Generator

ticks = 4

CLK

out

Counter:

output: Counter == 0, alternatively ~| counter

Next State: (counter == ticks) ? 0 : counter + 1
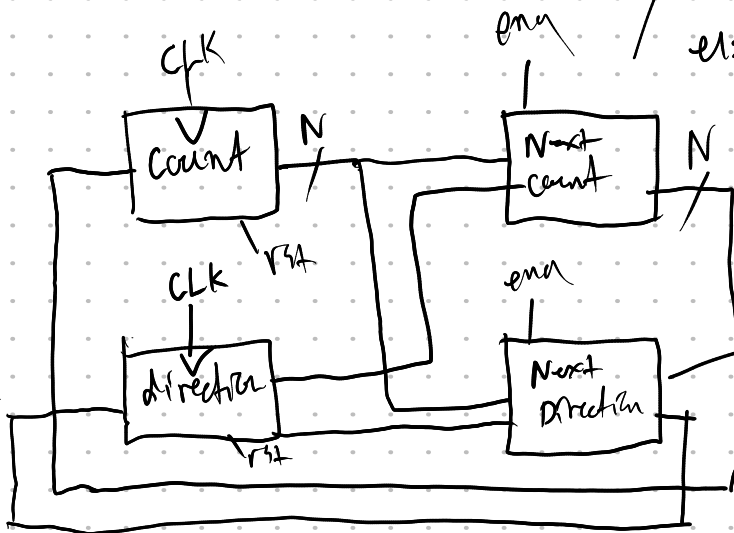
Triangle Wave

direction : up
            down

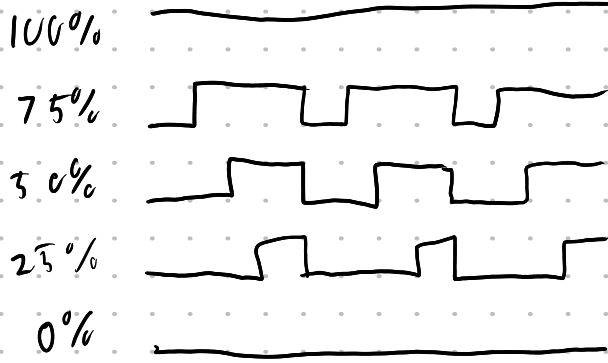count : $[N-1:0]$

```
if ena
  case Direction
    up: count +1
    dwn: count -1
else count
```
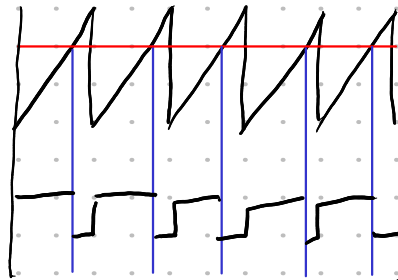


```
if ena
  case Count
    111..110    down
    000..001    up
    default     Direction
```

PWM Generator

100%

75%

50%

25%

0%

Counter , N bit bus

Out

Binary value of duty input

Out = Counter < duty

↳ make $2^n - 1$ always on

∟ out = (counter < duty) | (duty == -1)

clk

duty   ena

N count

Counter

Step

out