

Title:

History Of The Computerflip-Flops - A Basic Counter

Word Count:

811

Summary:

Flip-Flops - A basic counter

We looked at the Binary system, and basic computer logic elements, in previous articles, "It's a binary world - how computers count" and "How computers add - a logical approach".

Now we can combine two parts of these articles to look at a counter. Another common logic element in a computer is a counter or timer. This can be used to count items going past a sensor on an assembly line, or possibly a count-down timer. For example, if you have a late ...

Keywords:

history of the computer, computer history

Article Body:

Flip-Flops - A basic counter

We looked at the Binary system, and basic computer logic elements, in previous articles, "It's a binary world - how computers count" and "How computers add - a logical approach".

Now we can combine two parts of these articles to look at a counter. Another common logic element in a computer is a counter or timer. This can be used to count items going past a sensor on an assembly line, or possibly a count-down timer. For example, if you have a late model washing machine it will have a simple computer using a count down timer to give 10 minute wash cycle, etc.

There are several types of counter, nearly all of which use a basic element of electronics, the Flip-Flop. And you thought they were rubber shoes English people wear to the shower or the beach. (At this point Australians say "I thought they were called thongs").

OK back on topic. The flip-flop is as old as electronics, and is a classic example of the binary system. It has two possible stable states, A or B, and can

be 'toggled' from one state to the other, just like a 'push-on, push-off' switch. It was originally made with two vacuum tubes (or one, for example a double triode).

It normally has two outputs, one being the complement of the other. That is, if one output(A) is a logic 0, the other(B) is a logic 1, and vice-versa. The input, or Toggle(T) is at logic 0 until a pulse from a sensor, for example, comes along. This pulse takes the logic state to 1, then back to 0. The toggle effect, causing the Flip-Flop to flip, is actually the CHANGE from 0 to 1.

In logic terms the flip-flop is made up using AND and OR gates, in logic circuitry it is just a 'black box' labelled FF. Several FFs may be grouped into yet another black box, a counter, timer, or multivibrator.

We can make up a Truth Table, which we have used before. If you recall, a truth table tells you what the Output will be for all possible Inputs.

TRUTH TABLE for Flip Flop - Toggle (C)hange,- Outputs A and B.

INITIAL STATE

T B A

0 1 0 'A' output is 0

PULSE #1

T B A

C 0 1 'A' output is 1

PULSE #2

T B A

C 1 0 'A' output is 0

Now we string some flip-flops together to make a counter. Say we have a sensor on a beer bottling machine, which has to count 5 bottles before switching the feed, we need to count up to 5, or 101 in Binary. We will need 3 flip-flops, for binary bits 0,1 and 2, corresponding to decimal bit value of 1,2 and 4.

We will take the A output of the 3 flip-flops to a decoder black box, which we can use to detect when we get to 5, then switch the feed. The B output of flip-flop 0 is passed to the toggle input of flip-flop 1 via an AND gate, so the next pulse from the sensor (which goes to all 3 flip-flops) at this AND gate will toggle the flip-flop, depending on the value of the B output, 0 or 1. Similarly the B output of flip-flop 1 goes to the toggle of flip-flop 3 via an AND gate.

Our 3 Flip-Flops now come up with a truth table like this:-

INITIAL STATE

FF2 FF1 FF0

TBA TBA TBA

010 010 010 'A' outputs 000 - 0

PULSE #1

FF2 FF1 FF0

TBA TBA TBA

C10 C10 C01 'A' outputs 001 - 1

[The (C)hange flips FF0 (always). FF1 & FF2 are blocked by the AND gate which needs a 0 input from the previous FF 'B' output AND the pulse change.]

PULSE #2

FF2 FF1 FF0

TBA TBA TBA

C10 C01 C10 'A' outputs 010 - 2

[The (C)hange flips FF0 (always). FF1 flips because the 'B' output from FF0 is a 0 when the Pulse arrives. FF2 is blocked as before.]

PULSE #3

FF2 FF1 FF0

TBA TBA TBA

C10 C01 C01 'A' outputs 011 - 3

[FF0 flips, FF1 is blocked again, as is FF2.]

PULSE #4

FF2 FF1 FF0

TBA TBA TBA

C01 C10 C10 'A' outputs 100 - 4

(FF0 flips, FF1 flips, FF2 flips.)

PULSE #5

FF2 FF1 FF0

TBA TBA TBA

C01 C10 C01 'A' outputs 101 - 5 count complete!

[FF0 flips, FF1 and FF2 are blocked.]

This counter can count up to 111, 7 decimal, it then resets to 0. A couple of

interesting points to note are:-

1. FF0 flips every pulse. FF1 flips every 2 pulses. FF2 flips every 4 pulses etc. These facts can be used to make up a divider, which can be cascaded. For example the 4 pulse output can go to a second counter which also gives a 4 pulse output, totalling 16. This can be expanded to make up a decadic counter by decoding a count of 1010 (10 decimal) and using this to toggle the next counter, etc. What about 60 and 12 for your digital watch?

2. Look at the 'B' outputs from the counter. In sequence the values are:- 111, 110, 101, 100, 011, 010 (7,6,5,4,3,2 decimal). See the pattern? That's right - a countdown timer! We'll be using this in a later article.