Successive Approximation Register (SAR)

How do we get to digital from analog? The SAR is one way.

Let's decode the decimal number 170 to its equivalent in binary in 9 bits. 170 is smaller than 256, so get a 0 there. We can with 128 (170 – 128 = 42). Not with 64. Yes with 32, no with 16, yes with 8, no with 4, yes with 2, no with1.

Get 010101010 from 128 + 32 + 8 + 2.

Initially the SAR will output a digital value equivalent to half the full scale voltage (the MSB (most significant bit)). This output will be converted by the DAC (digital-analog converter) into a voltage that forms one of the inputs to the comparator. If the unknown voltage, connected to the alternate input of the comparator, is higher than current DAC output, then the MSB is maintained and the next significant bit is then set and this is weighed against the unknown input. This process continues until the LSB is reached.

Diagram 1 of an ADC. The ADC has a DAC within it. We apply the 4-bit value 10 to the DAC, and then move it to the comparator, which compares it to 10. Keep it if lower than or equal to our number.

If change the 4-bit value to 12, the 12 is higher than the 10 coming in.

If compare with 10, keep the 1. If compare with 11, make it a zero. See logic below diagram 1.

Large Scale Integration chip does this; it is an analog-to-digital converter. Has counter, comparator, and digital-to-analog converter. Clock is external. The comparator is an analog comparator. A lot easier to convert D to A than A to D.

Advantages

Much faster results. DAC output converges on analog signal input in much larger steps than with the 0 to full count sequence of a regular counter.

The updates for this ADC occur at regular intervals, unlike the digital ramp ADC circuit.

Result is always n clock pulses after the start.

What Makes Up a SAR?

Includes:

- Sample/hold control circuit
- Comparator
- SAR
- DAC (digital to analog converter)
- output latch

Sample/hold circuit is at the front. Sample level of the analog wave where ever we are on the waveform. Want it to make the sample and to hold that value while we process it.

Comparator to compare the held value with the one at the output of the DAC.

For n bits of resolution, there are n comparator operations, each stored in the register. For example, for 8 bits of resolution, execute 8 comparisons on each input signal (sample).

For 601, clock fed to the SAR would run at 135 MHz (10 bits at a rate of 13.5 MHz for luminance).

Sample/Hold Control Circuit

This circuit is an interface between the ADC's input and the rest of the ADC. What is the one device that holds the *analog* voltage sample? A capacitor. Have a switch tied to a capacitor that only closes during the sample time. The switch is usually a transistor. Pulse drives the transistor at the point at which we want to sample the wave. For 601, clock runs at 13.5 MHz for the sampler. Precision capacitor so that it will quickly charge and discharge at the sampling frequency. Involves the RC time constant, the remainder of the circuit being the R.

The sample/hold control circuit sustains a certain point of an input signal for an amount of time necessary for the rest of the circuit to react to its value. The S/H circuit is a pair of voltage follower op-amps linked by a FET switch and a holding capacitor. Voltage follower = emitter follower = common collector.

Limitations:

- acquisition time (how fast it can generate the digital output)
- droop rate (how capacitor discharges)
- sample-to-hold settling time (how fast I can sample this and get the actual voltage)
- slew rate (related to the sharpness (rate of change) of the wave being sampled)

just to name a few.

Schematic of Sample and Hold circuit in Diagram 2. Value of C is a compromise. Big C gives slow acquisition, small C gives too much drift (due to discharge through the circuitry).

The value of the capacitor is the most important thing in the S/H circuit.

Comparator

Used in many types of A/D converters. Is the simplest interface from an analog signal to a digital signal (if same is yes, if not same is no). Compares two voltage values on its inputs. If the voltage on the + input is greater than the voltage on the – input, the output will be a logic high. Else the output will be a logic low (less than or equal to).

Where have we seen a comparator? In a PLL (the phase comparator)? What did we compare? The input signal and the feedback (the output of the VCO).

Diagram 3.

Priority Encoder 74148

An 8-line to 3-line priority encoder. A priority is assigned to each input, when two or more inputs are simultaneously active, the input with the highest priority is represented at the output. Inputs I bar 0 – I

bar 7, outputs A bar 0 – A bar 2. E1 bar is the enable line. I bar 7 is the highest priority; if it is low, it will set A0, A1 and A2 to low, regardless of the values at any of the other I bar inputs. The outputs are in negative logic; here, for I bar 7 being low, the A's are all low, and interpreting them all as high, and adding them together with their bit level, you get 4 + 2 + 1 = 7.

How to know if we have inverted logic? The bars above the inputs and outputs is one indication.

Why use inverted logic here? Look at an example of a car door light. Have a dome switch and a door light. When you open the door, the switch closes and turns on the light. How is this wired? Apply a ground to activate the light. One side of the switch is connected to ground. Diagram 4.

Priority Encoder with an input from a Comparator

Diagram 5.

Diode logic. Had diode-transistor logic before TTL, then before that had diode logic. Using diodes as logic gates. Sends around a PC board with a bunch of diodes in them (it's a switch board from an RTS IFB(?) unit). The board was probably developed before TTL. Comparators were expensive at the time this circuit was developed, so built their own comparators.

This circuit is more complex, but we don't have to do the counting thing.

Projects