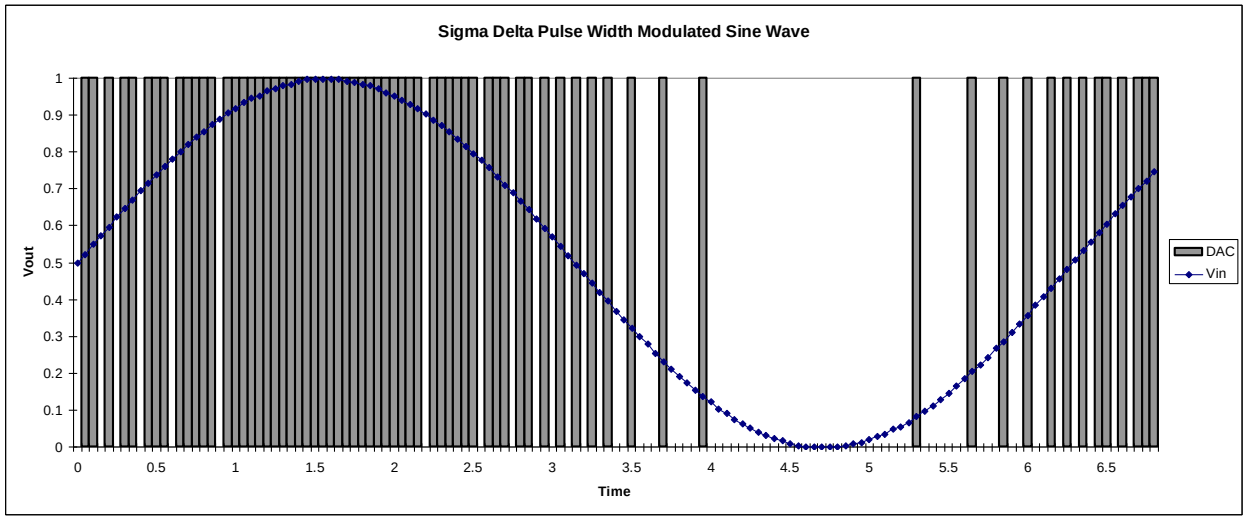
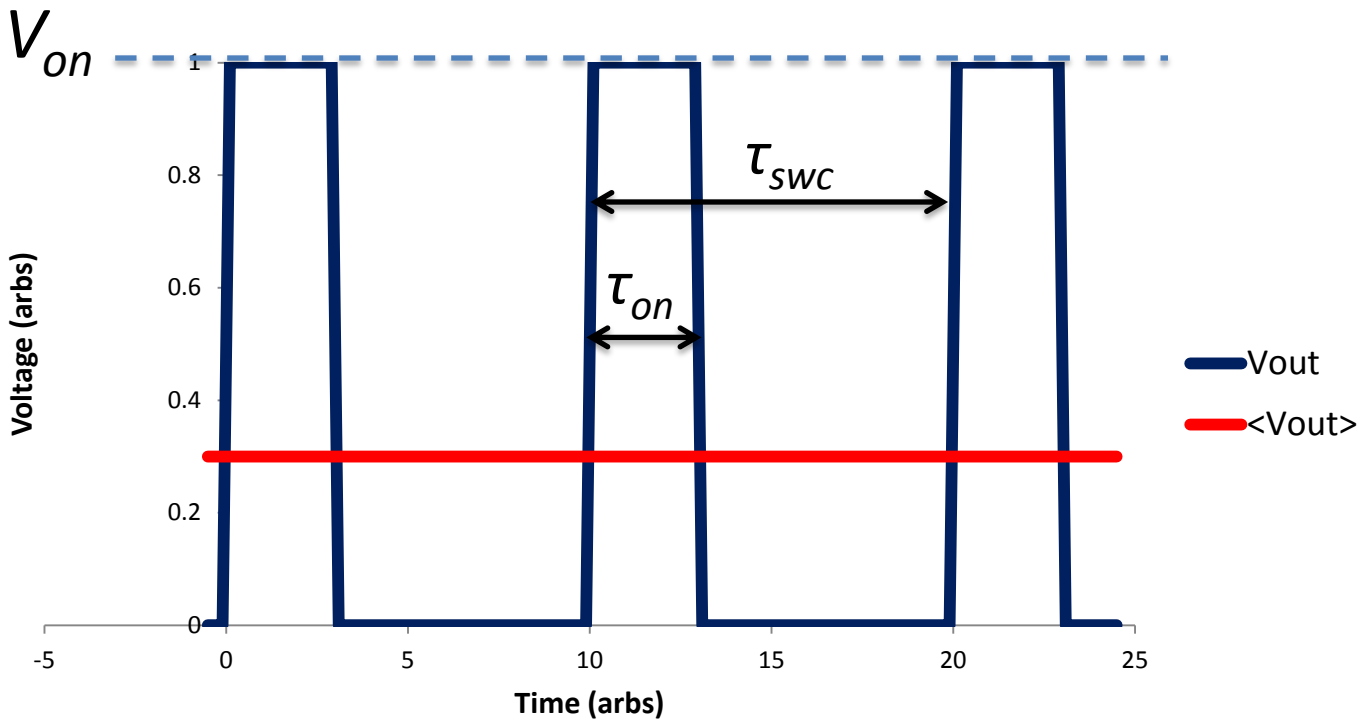
Pulse Width Modulation & FPGA



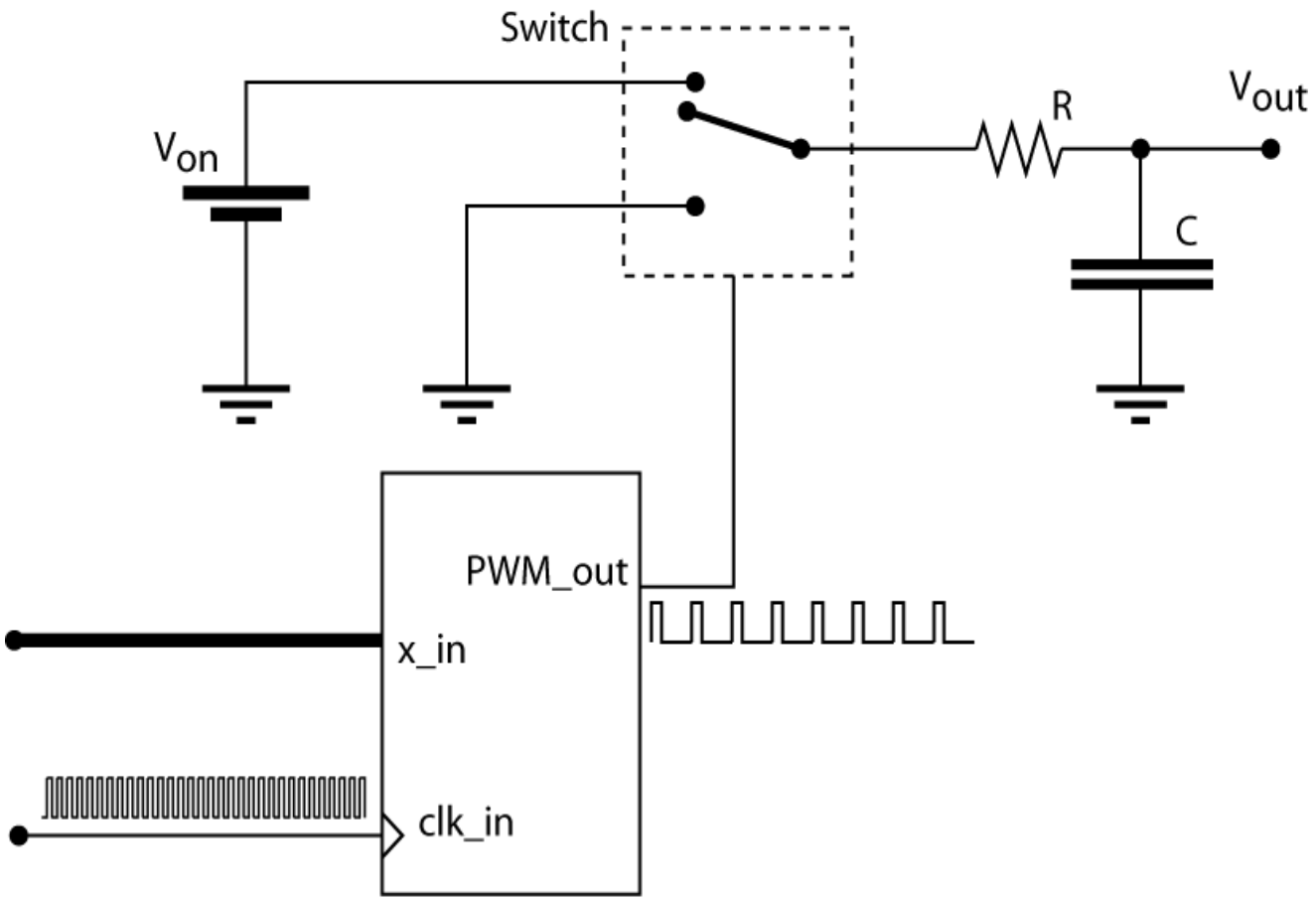
BFY
AAPT Philadelphia 2012
Kurt Wick (wick@umn.edu)
University of Minnesota

PWM Theory

$$\langle V_{out} \rangle = \frac{\tau_{on}}{\tau_{swc}} V_{on}$$



PWM Circuit Components

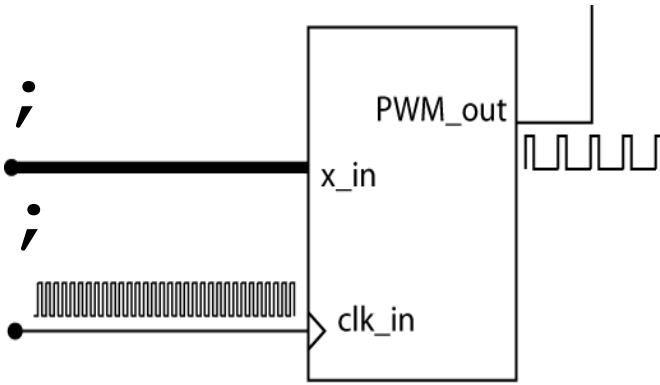


Switch Control Algorithm,
i.e., PWM Algorithm

Simple PWM Control Algorithm

- *An n-bit counter continuously increments from 0 to its maximum value, i.e., 2^n-1 and then repeats the cycle.*
- *Range of input value x_in: $0 \le x\_in \le 2^n-1$*

```
if ( counter < x_in )
        PWM_out <= 1;
    else
        PWM_out <= 0;
counter <= counter+1;
```



Switch Control Algorithm, i.e., PWM Algorithm

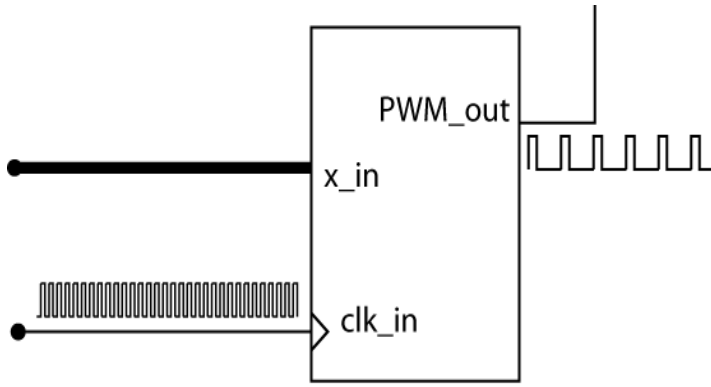
Verilog Implementation of Simple 8 Bit PWM Algorithm

```
module SimplePWM(clk_in, x_in, PWM_out);

        input clk_in;   //clock for counter
        input [7:0] x_in;               //control value that
                                        //defines pulse width
        output reg PWM_out = 1;         //PWM signal out

        reg [7:0] counter = 0;

        always@ (posedge clk_in )begin
                if ( counter < x_in )
                        PWM_out <= 1;
                else
                        PWM_out <= 0;
                counter <= counter+1;
                end
endmodule
```



Switch Control Algorithm,
i.e., PWM Algorithm

A2D Key Concepts: **Resolution / Sensitivity**

- Resolution of an n-bit PWM A2D is: $\mathbf{V_{on} / 2^n}$
- (Hypothetical) Resolution for our BASYS board PWM A2D with V_{on} = 5 Volts would be:

bits	Resolution (Volts)
8	1.9E-02
16	7.6E-05
32	1.2E-09
64	2.7E-19

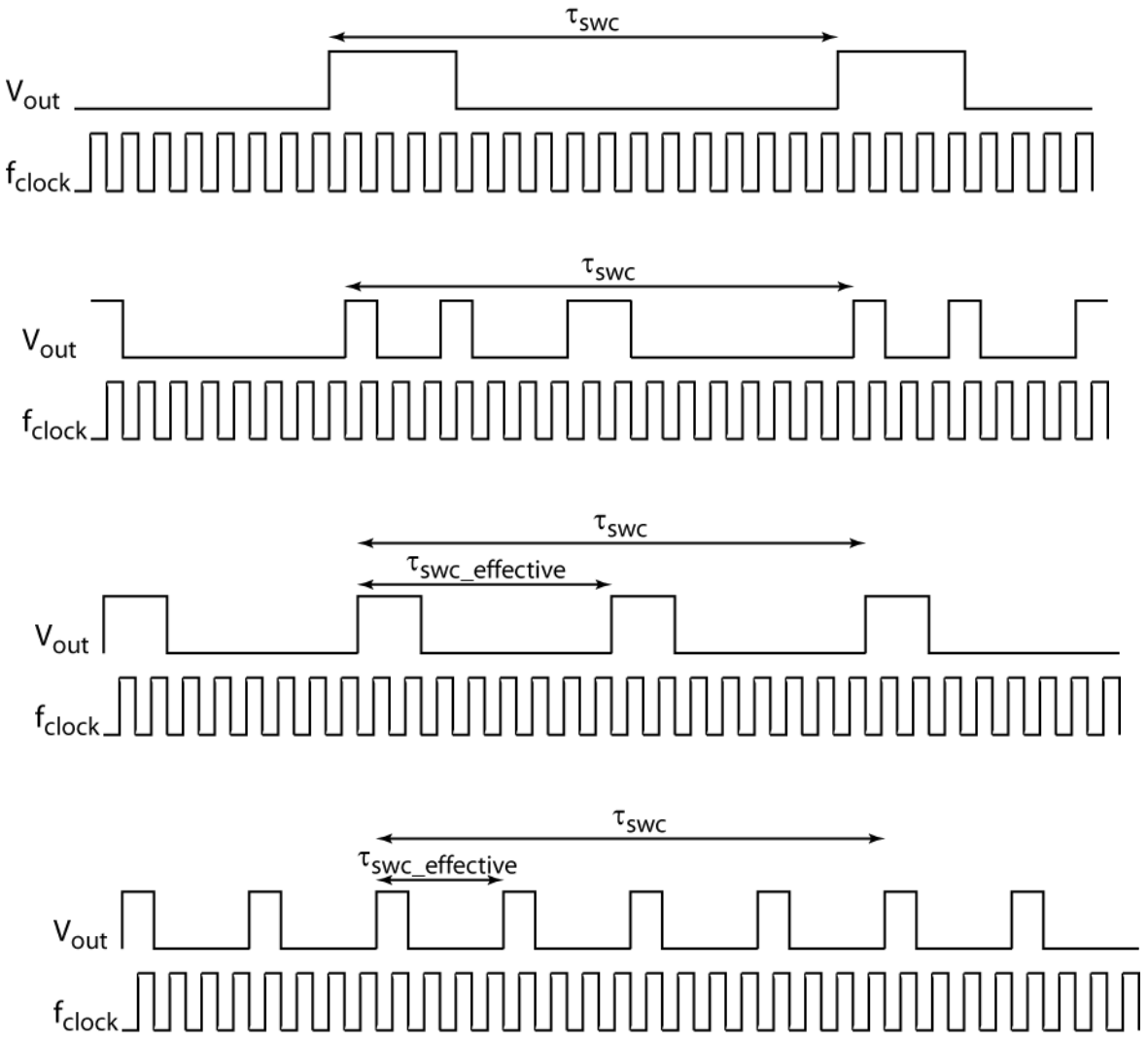
A2D Key Concepts:
Conversion Time

- It takes <u>at least</u> one complete counter cycle to average V_{out}. For a counter running at f_o, this corresponds to: $\mathbf{2^n/f_o}$
- (Optimal) Conversion Time for our BASYS board PWM A2D with f_o = 25 MHz:

Bits	Resolution (Volts)	Conversion Time
8	1.9E-02	10.2 usec
16	7.6E-05	2.6 msec
32	1.2E-09	2.8 min
64	2.7E-19	23000 years

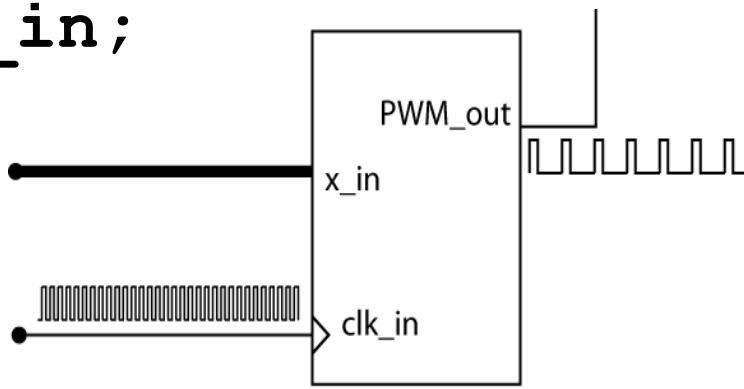
Sigma Delta Concepts

Though $V_{out}(t)$ is different in each of these timing diagrams, $\langle V_{out} \rangle$ remains identical.



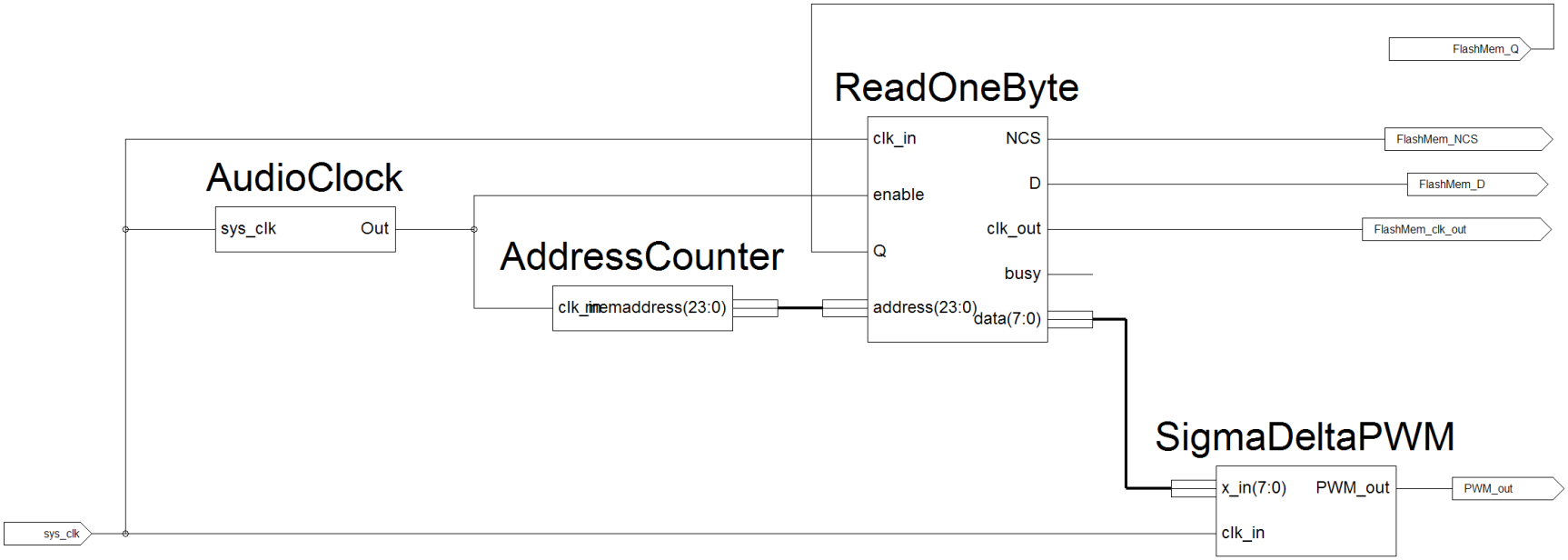
Sigma Delta PWM Algorithm

```
if ( Sigma >= Delta )
     {
     Sigma = (Sigma - Delta) + x_in;
     Out = 1;
     }
else
     {
     Sigma = Sigma + x_in;
     Out = 0;
     }
```



Switch Control Algorithm,
i.e., PWM Algorithm

PWM Application: Music Player



Conclusions about PWM

- Pulse Width Modulation can be used to create an analog signal from a digital signal.
- Allows the reduction of a DC signal while being much more energy efficient than, for example, a passive voltage divider.
- Sigma Delta algorithm can also be used for voltage to frequency conversion.

Educational Goals

- The PWM exercises expose students to basic digital concepts such as clocks and counters.
- They are easily implemented with an FPGA and thereby exposes students with this ubiquitous electronic component.
- Familiarizes them with digital-to-analog converters and the basic concepts of resolution and conversion time.
- The exercise can be extended by turning it into an analog-to-digital converter using successive approximation and a state machine.