

Pulse Width Modulation Generator

--- ECE 734 Final Project Proposal

Qian Zhang Nov 1 2000

Background Introduction:

Modulation:

There are many forms of modulation used for communicating information. When a high frequency signal has amplitude varied in response to a lower frequency signal we have AM (amplitude modulation). When the signal frequency is varied in response to the modulating signal we have FM (frequency modulation). These signals are used for radio modulation because the high frequency carrier signal is needed for efficient radiation of the signal. When communication by pulses was introduced, the amplitude, frequency and pulse width become possible modulation options. In many power electronic converters where the output voltage can be one of two values the only option is modulation of average conduction time.

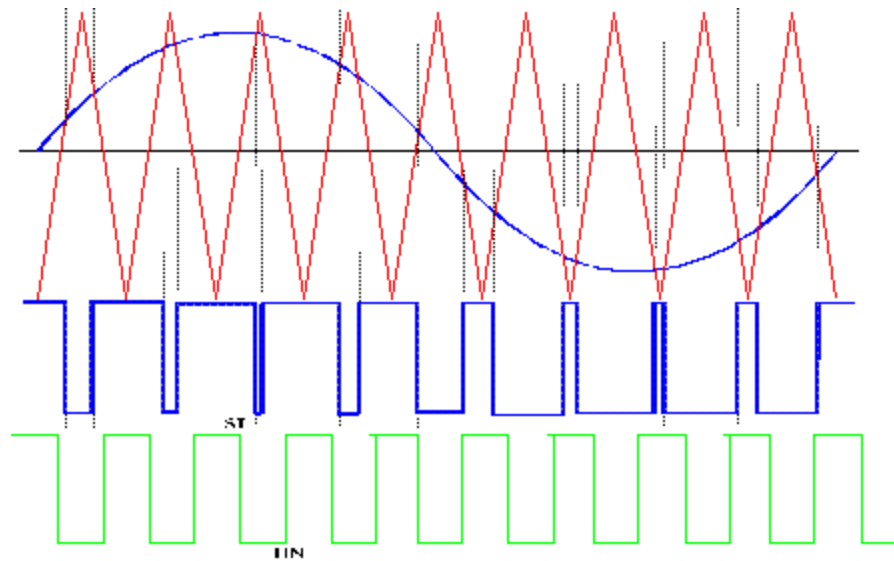


FIG PWM_1 Unmodulated, sine modulated pulses

There are many kinds of modulations, in this project, we will mainly concentrate on linear modulation

Linear Modulation

The simplest modulation to interpret is where the average ON time of the pulses varies proportionally with the modulating signal. The advantage of linear processing for this application lies in the ease of de-modulation. This kind of PWM is the one we will try to implement in this project.

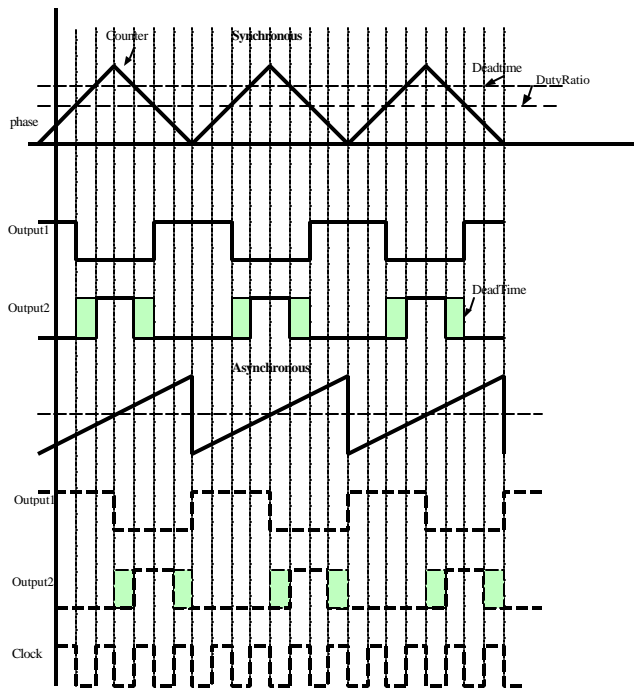


Fig Linear Pulse Width Modulation

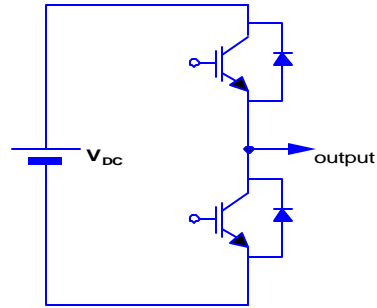


Fig A general Power electronics circuit

Fig Linear Pulse Width Modulation shows us the general-purpose pwm waveform, When upper IGBT/MOSFET turns on(output 1), the lower turns off. To avoid short circuit, the two IGBT/MOSEFT could not be turn on at the same time, so here deadtime is used.

PWM Generator

Generally speaking, the PWM generator is to generate the pulse as the control signal of IGBT to let it turn on or turn off.

- Inputs:
 - Reset --- clears the PWM to initial state
 - SWITCHING_F --- Switching Frequency from the DSP
 - DUTYRATIO --- Dutyratio from the DSP
 - DEADTIME --- Deadtime from DSP
 - MINDEADTIME --- Even though dead time should come from DSP, however, to protect the circuit, and we need to setup a mindeadtime at RESET stage, when Input DEADTIME is smaller than the MINDEADTIME, we need to use MINDEADTIME.
 - SAMPLING_F --- Sampling frequency from DSP
 - CONTROL ---
 - when Control = 1 asynchronous wave generator
 - when Control = 0 synchronous wave generator
 - PHASE --- The initial phase of the PWM waveform

- Outputs:
 - WAVE1 --- Output waveform
 - WAVE2 --- Output waveform
 - CONV_START --- A signal to tell convert start

Others Important:

- 1) The SWITCHING_F, DEADTIME, DEADTIME, CONTROL ,PHASE signals can be only changed when one period PWM finish.

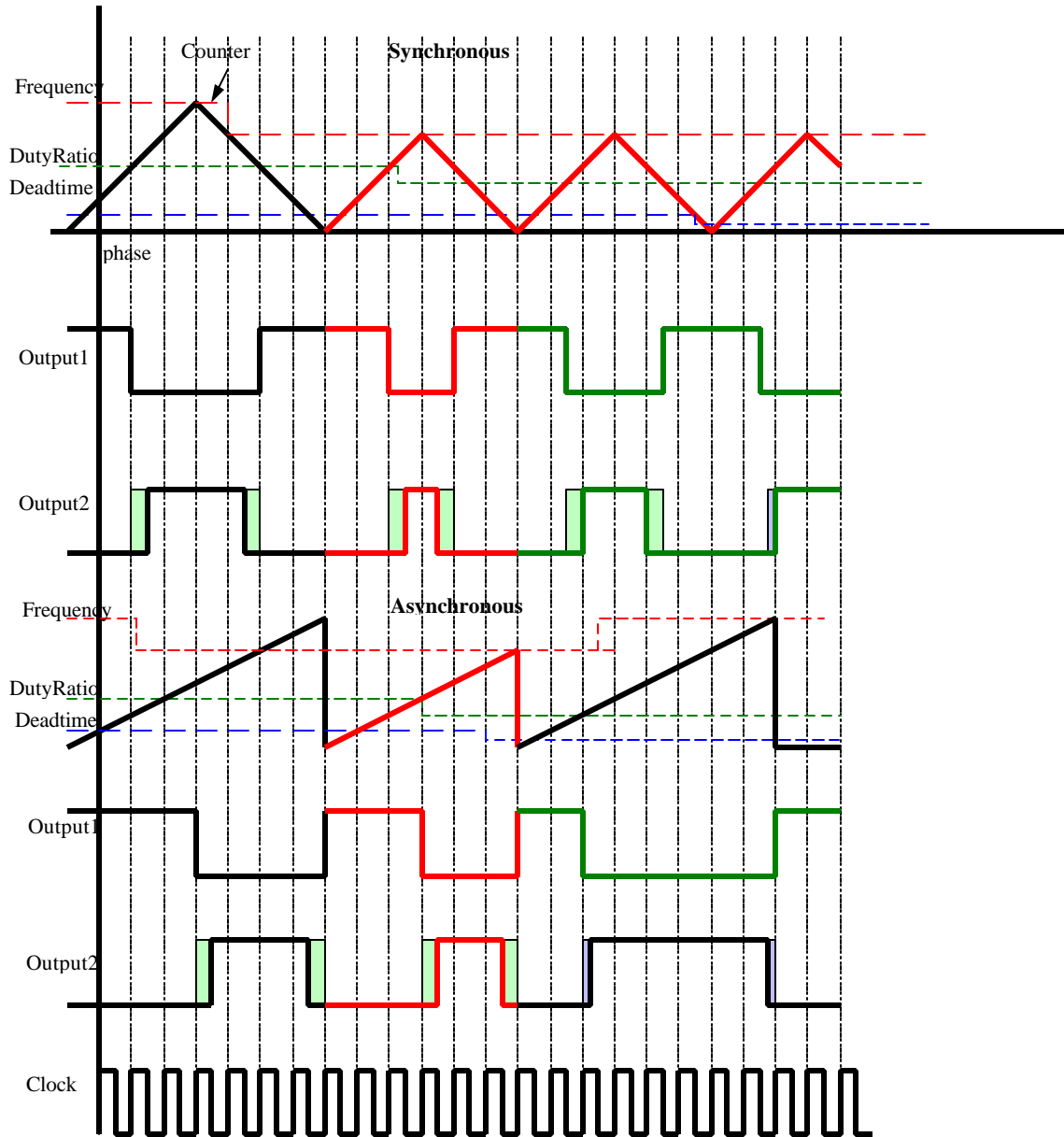


Fig Linear Pulse Width Modulation

2) DEADTIME Protection: The two output cannot be conducted at ANY time, so even if the input of deadtime is zero, the output can't be zero, so here we add input of MINDEADTIME, this can only be changed at RESET time.

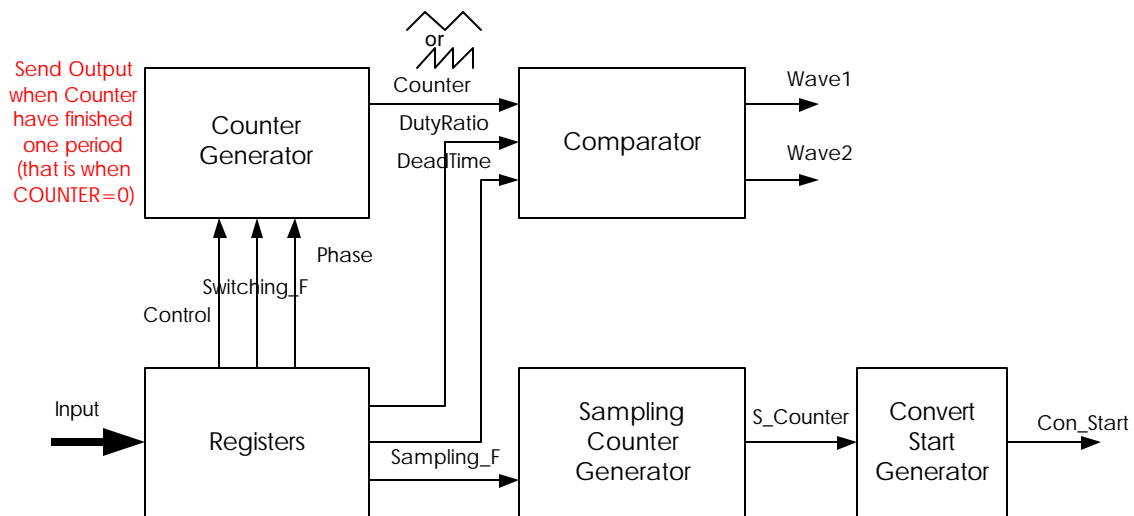
3) PHASE change.

For Synchronous Way, the effective input phase can only be in $[0, 2\text{switching_f}]$;

For Asynchronous Way, the effective input phase can only be in $[0, \text{switching_f}]$;

Phase input at any other range has no use.

Block Diagram



PWM Generator, should have at least 6 channels, the detailed description of my channel is explained below.

I will plan to implement PWM with VHDL first and write test bench to do the simulation.

If possible, we will also try to synthesis, implement and download our design to the FPGA 4000C series. In this way, we may demo some results in the hardware