# Experiment 3 - Function Generator

Mahdi Vajhi, Seyed Alireza Mirshafiee
810101558, 810101532

***Abstract*— The purpose of this test is to implement a function generator that can generate sine, square, etc. waves. The implementation of the sine wave is by reading the first 1/4 of it from RAM and making the rest of the wave using it. The rest of the waves are implemented using mathematical conditions and operations.**

***Keywords*— Function Generator, PWM, Frequency Selector, Amplitude Selector, Digital To Analog**

## I. Waveform Generator

### *A. Simulation Outcomes*

The simulation results of all types of waves are presented in Figure 1.

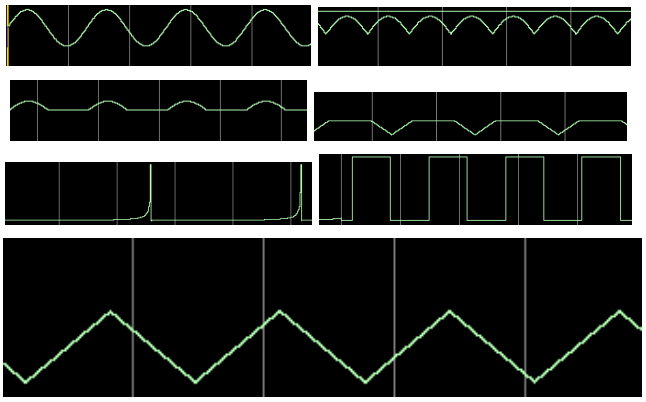
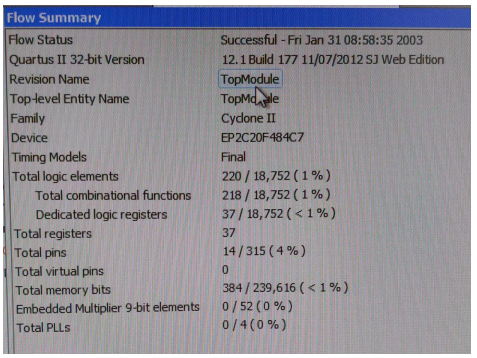


Fig. 1 Simulation Outcomes

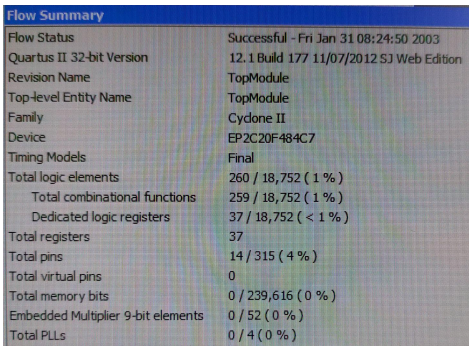
### *B. ROM*

The result of circuit compilation using ROM with keyword is shown in Figure 2 and without it in Figure 3. If we do not use the keyword, the data will be logically implemented instead of being placed in the ROMs of the board.



| Flow Summary | |
|---|---|
| Flow Status | Successful - Fri Jan 31 08:58:35 2003 |
| Quartus II 32-bit Version | 12.1 Build 177 11/07/2012 SJ Web Edition |
| Revision Name | TopModule |
| Top-level Entity Name | TopModule |
| Family | Cyclone II |
| Device | EP2C20F484C7 |
| Timing Models | Final |
| Total logic elements | 220 / 18,752 ( 1 % ) |
| Total combinational functions | 218 / 18,752 ( 1 % ) |
| Dedicated logic registers | 37 / 18,752 ( < 1 % ) |
| Total registers | 37 |
| Total pins | 14 / 315 ( 4 % ) |
| Total virtual pins | 0 |
| Total memory bits | 384 / 239,616 ( < 1 % ) |
| Embedded Multiplier 9-bit elements | 0 / 52 ( 0 % ) |
| Total PLLs | 0 / 4 ( 0 % ) |

Fig. 2 Compilation report for module using ROM with the romstyle keyword



| Flow Summary | |
|---|---|
| Flow Status | Successful - Fri Jan 31 08:24:50 2003 |
| Quartus II 32-bit Version | 12.1 Build 177 11/07/2012 SJ Web Edition |
| Revision Name | TopModule |
| Top-level Entity Name | TopModule |
| Family | Cyclone II |
| Device | EP2C20F484C7 |
| Timing Models | Final |
| Total logic elements | 260 / 18,752 ( 1 % ) |
| Total combinational functions | 259 / 18,752 ( 1 % ) |
| Dedicated logic registers | 37 / 18,752 ( < 1 % ) |
| Total registers | 37 |
| Total pins | 14 / 315 ( 4 % ) |
| Total virtual pins | 0 |
| Total memory bits | 0 / 239,616 ( 0 % ) |
| Embedded Multiplier 9-bit elements | 0 / 52 ( 0 % ) |
| Total PLLs | 0 / 4 ( 0 % ) |

Fig. 3 Compilation report for module using ROM without the romstyle keyword

## II. PWM

### *A. Explain PWM*

PWM is a module to convert digital data of our wave to analog wave. This element works in such a way that it has a 256 counter that compares in each circuit count whether the input value is greater than the counter or not. This subject causes more values of 1 to be placed on the output, and this causes the amplitude of the analog wave to increase.

### B. *Simulation Results*

As you can see in Figure 4, 5 and 6, PWM mode stays at 1 where the signal value is closer to 256, and this position on the analog output means more amplitude.

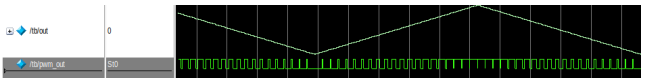


Fig.4 PWM for triangular wave

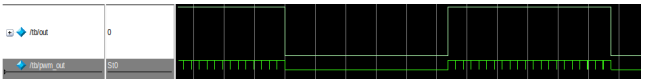


Fig.5 PWM for square wave

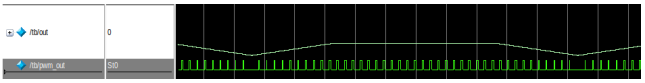


Fig.6 PWM for trapezius wave

## III. Frequency Selector

### A. *Simulation Results*

The results for different values are shown in Figure 7.

Fig.7 frequency selector simulation results

## IV. Amplitude Selector

### *A. Simulation Results*

We tested the circuit wave amplitude for different values and you can see the result in Figure 8.

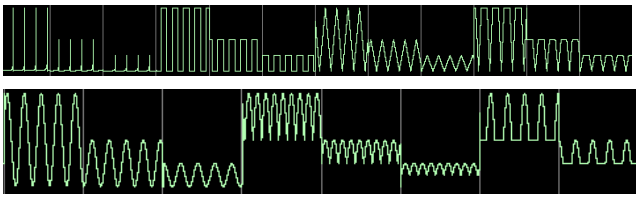


Fig.8 Simulation result for different values of wave amplitude

## V. Implementation

### A. *Schematic Diagram*

In Figure 9, you can see the connection schematic of the designed modules.

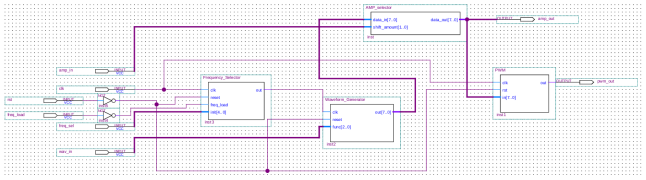


Fig.9 Schematic diagram of the Function Generator

### B. *Oscilloscope Visualization*

In Figure 10 to 13, you can see the effect of different amplitudes and frequencies on the sine wave, and in Figure 14, the waveforms of the remaining waves are shown.

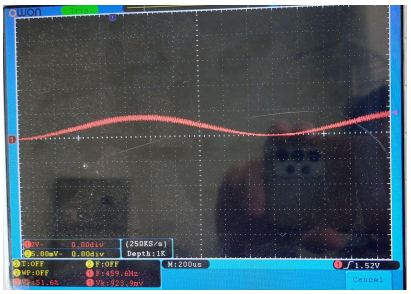


Fig.10 Sine wave with low amplitude and frequency

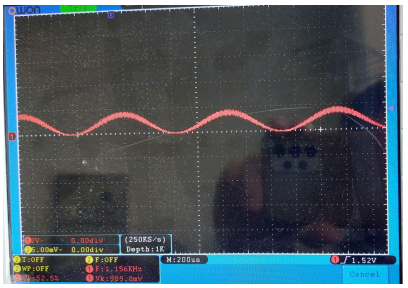


Fig.11 Sine wave with low amplitude

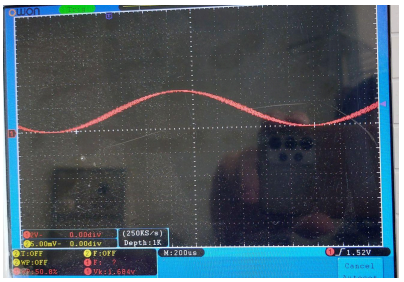


Fig.12 Sine wave with low frequency

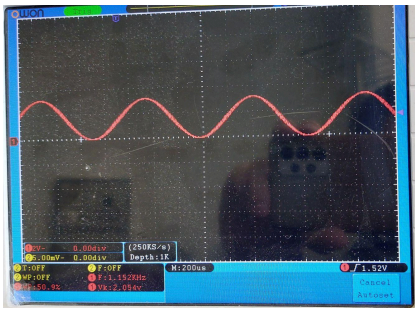


Fig.13 Normal Sine wave

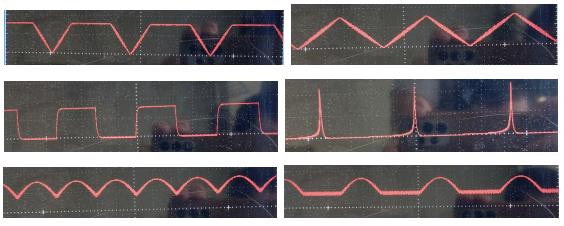


Fig.14 Other wave

## **VI.** Conclusions

In this experiment, We learned how to digitally produce a wave. We learned to convert analog signals to digital and finally we learned how to use ROM memory in an FPGA.

## References

[1] K.Basharkhah, Z.Jahanpeima “Laboratory Manual, Experiment 3,Sessions 6,7,8, Clock and Periodic Signal Generation V.spring 1403” University of Tehran, College of Engineering, School of Electrical & Computer Engineering