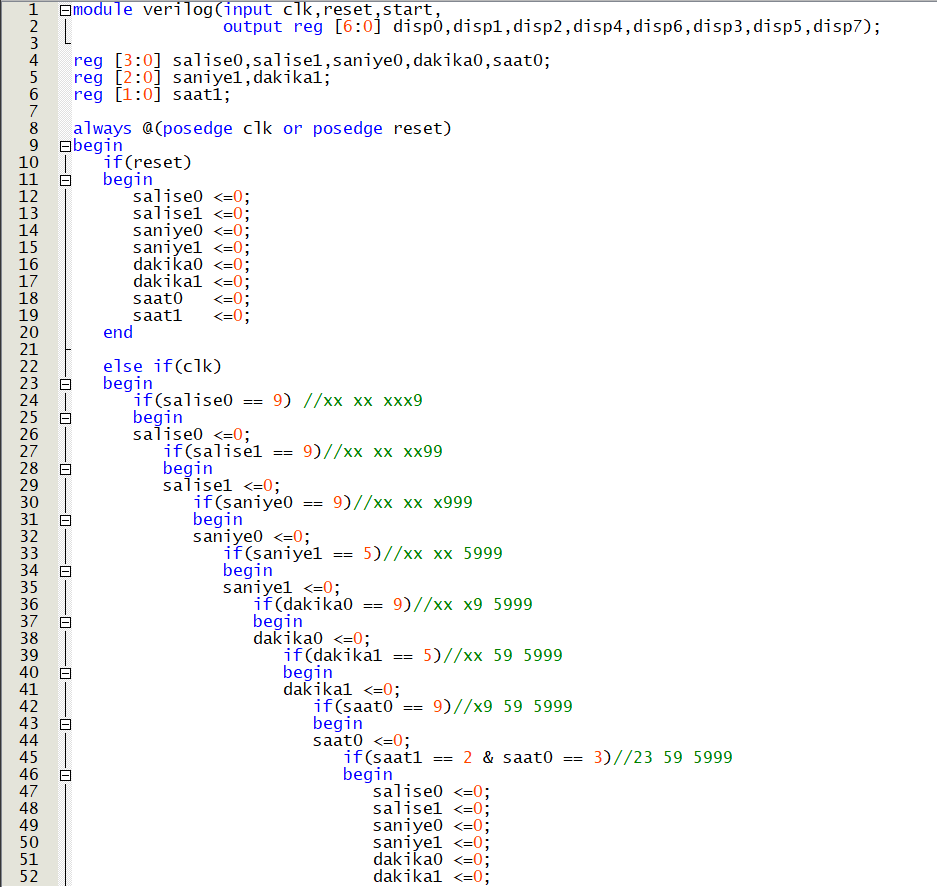
Workshop on Programmable Logic Devices

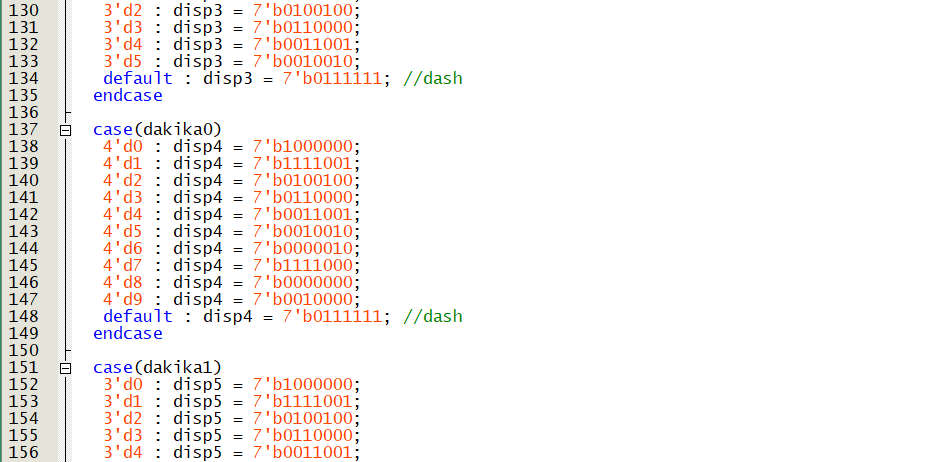
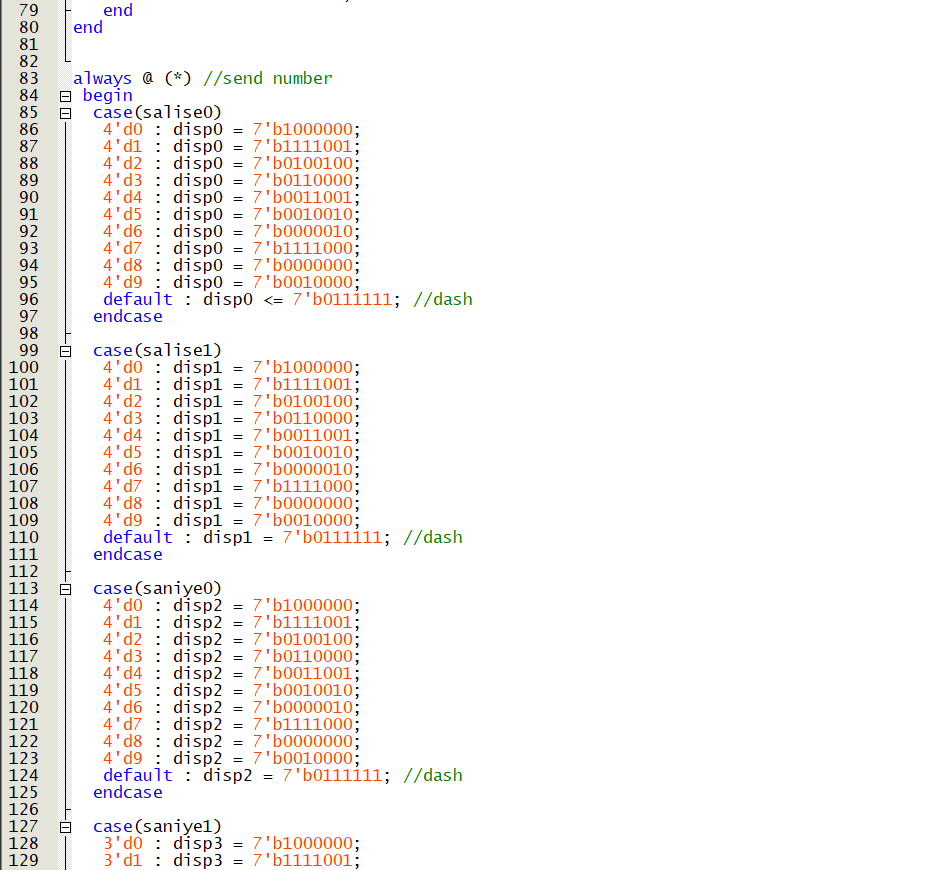
Łukasz Sajewski

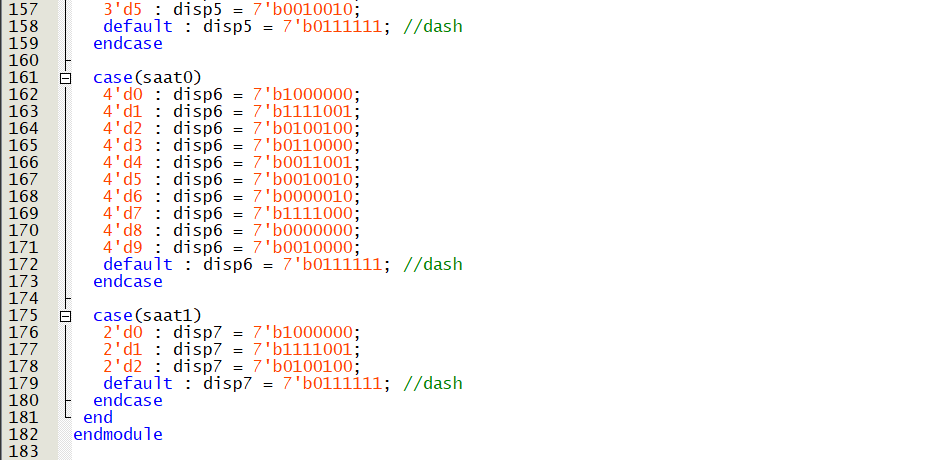
Tasks Part-3

Burak ELHAMAN

**Verilog Code**

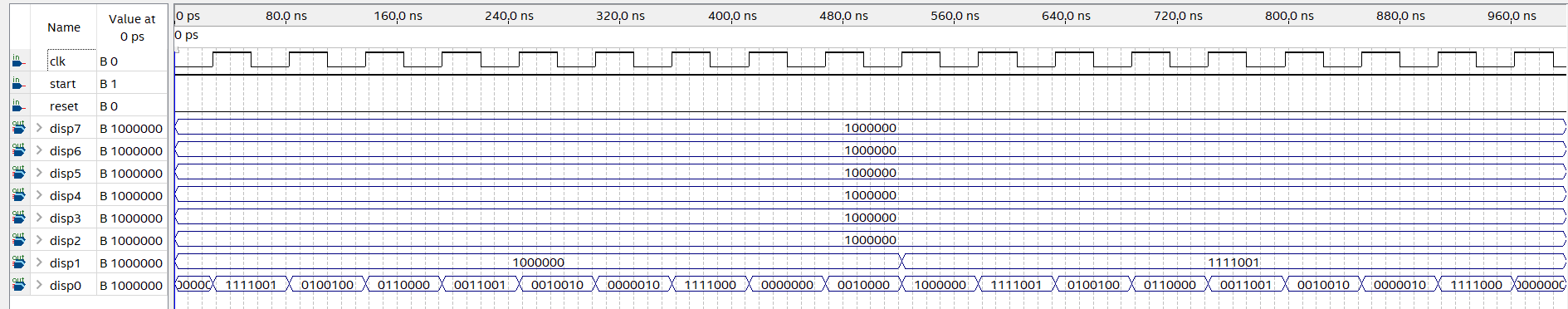






**Waveform**

We can see milisecond count in this picture.



Introduction

In this project I did clock and this clock has hour, minute, second, milisecond. Clocks has a big importance in our life. I used two different methods for doing this project. I learned these methods in FPGA course. First method is frequence converting, second method is Code to Block diagram converting.

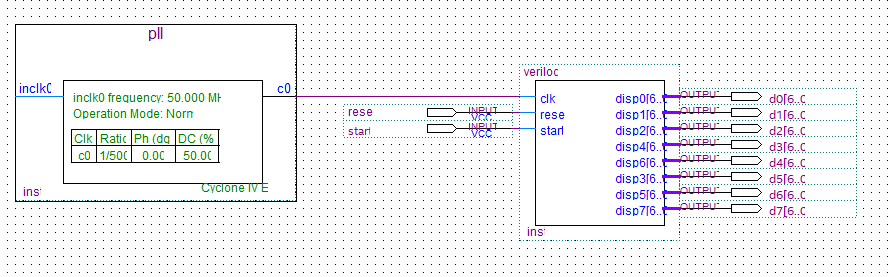
### Frequency Converting

Our normal clock input is 50 MHz. With this method we can get the clock output we want. Block Diagram clock input is 100 Hz. This is equal milisecond count. Ve can see stages in Fig-1 and Fig-2 and make settings. If we want, we can add poly output of the converter.

### Code To Block Diagram Converting

First we should write code after that we can convert. Our code is ready, we choose create symbol files for current file. Then we wait for creating after that its ready in symbol files. We can see stages in Fig-3 and Fig-4.

### Block Diagram



### Important Things about Verilog Language

We should add input and output in module brackets. Then we use reg for registration. We use always for main menu, if we put something in brackets it will depend on that or we put (\*) this symbol always running always. And every always running at the same time. If and case comments are similar with C language.

**Stages**

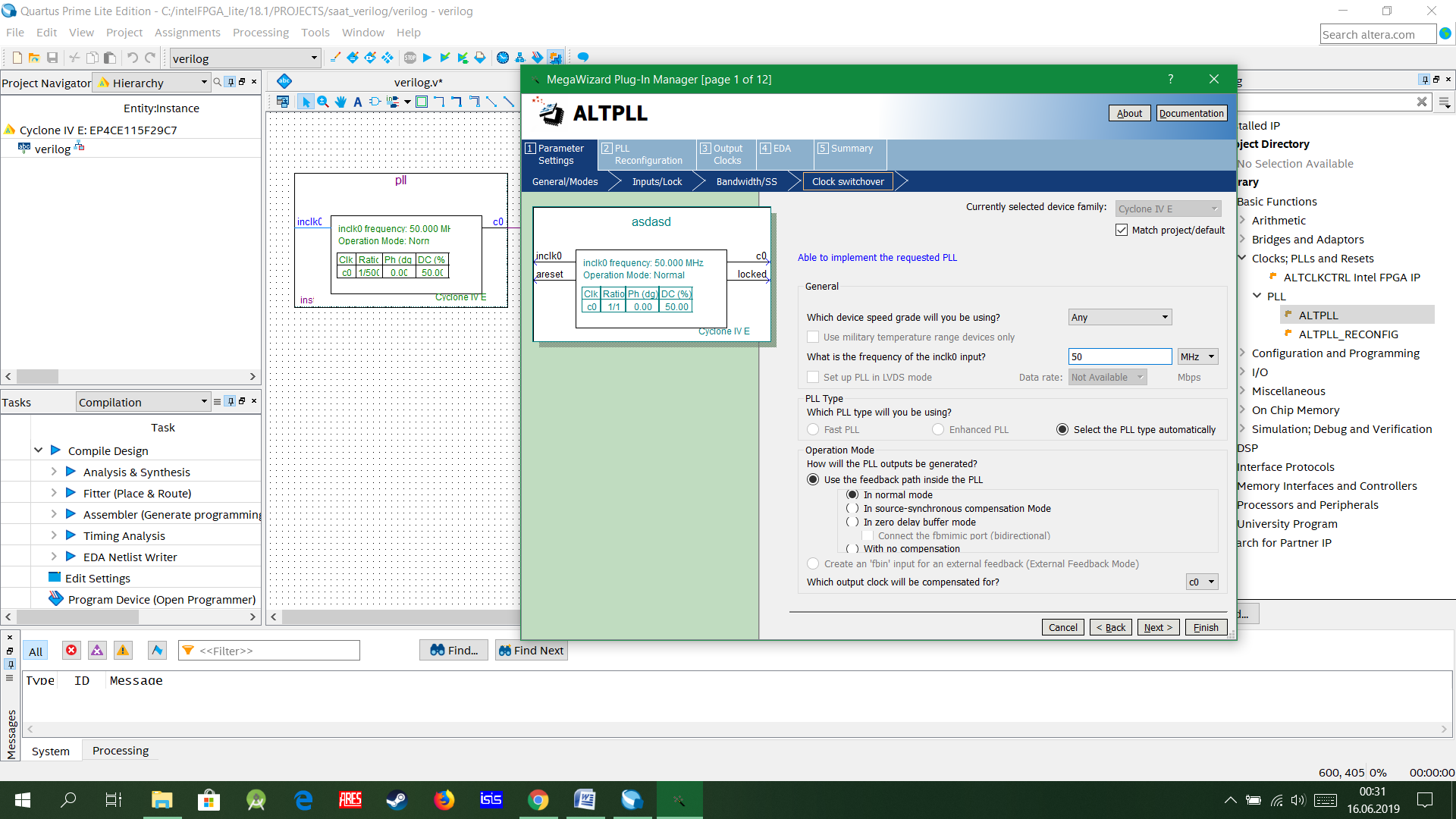
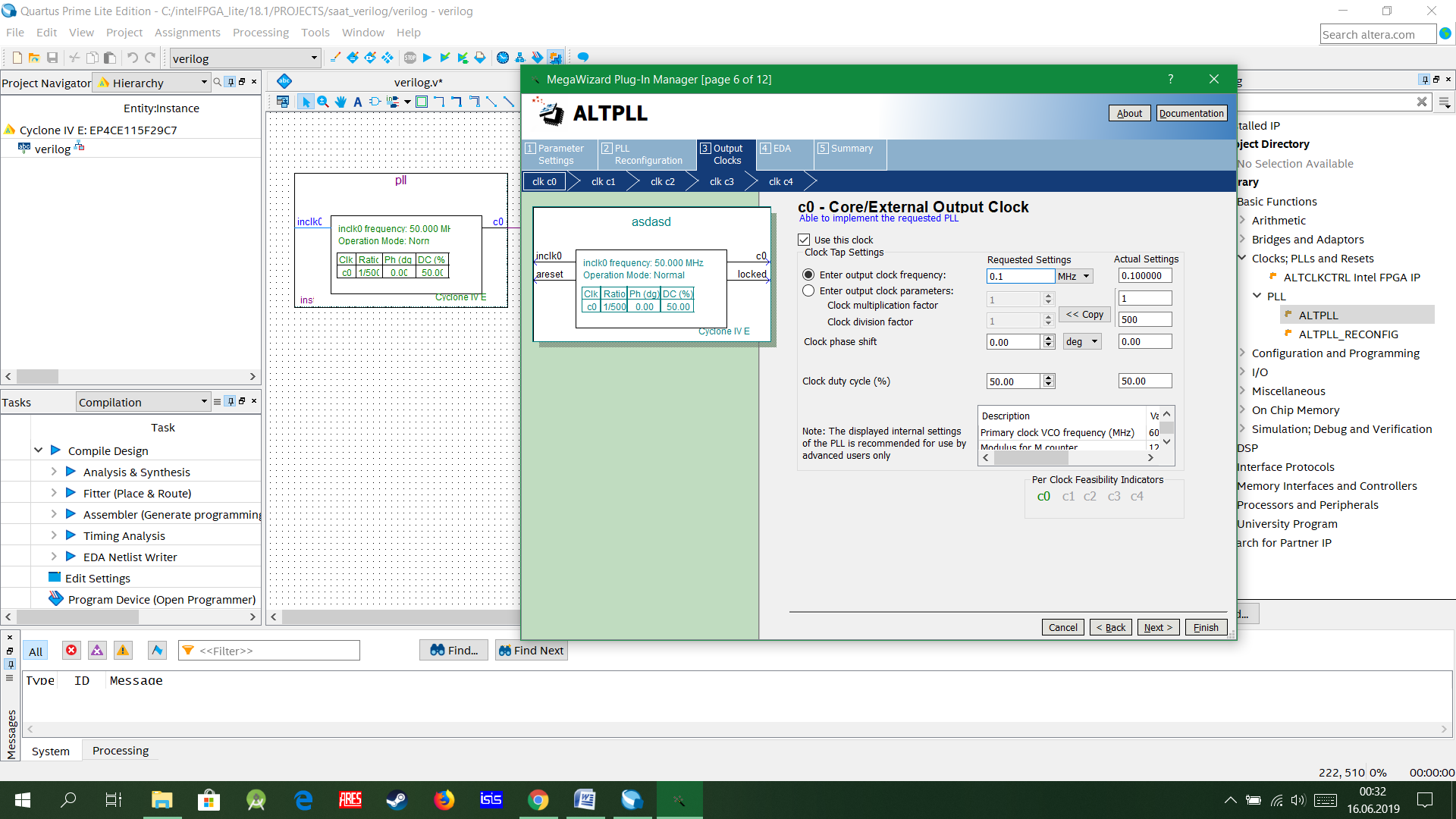


Fig-1 Frequance converting 50 MHz to 100 Hz



**Fig-2**

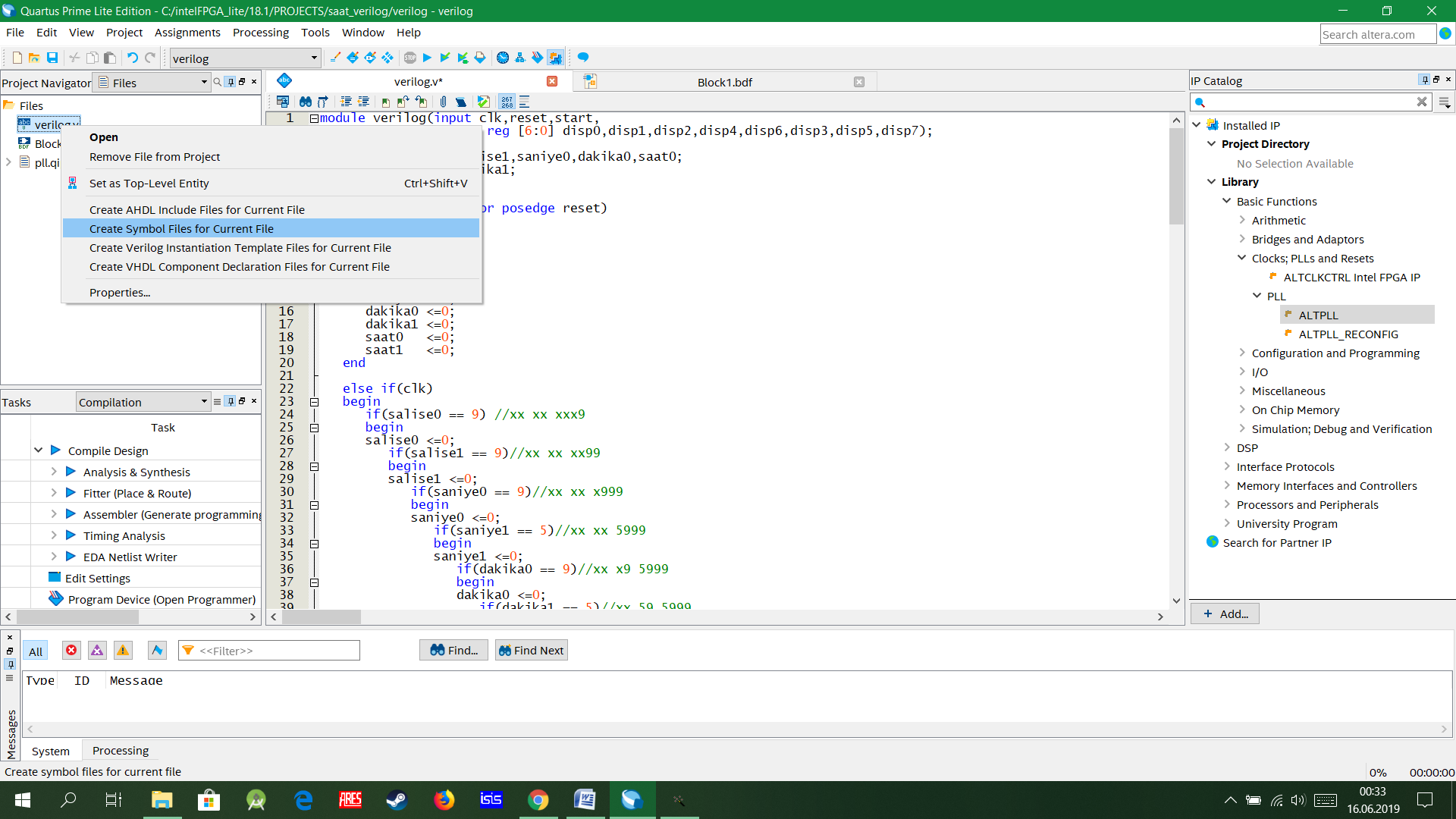
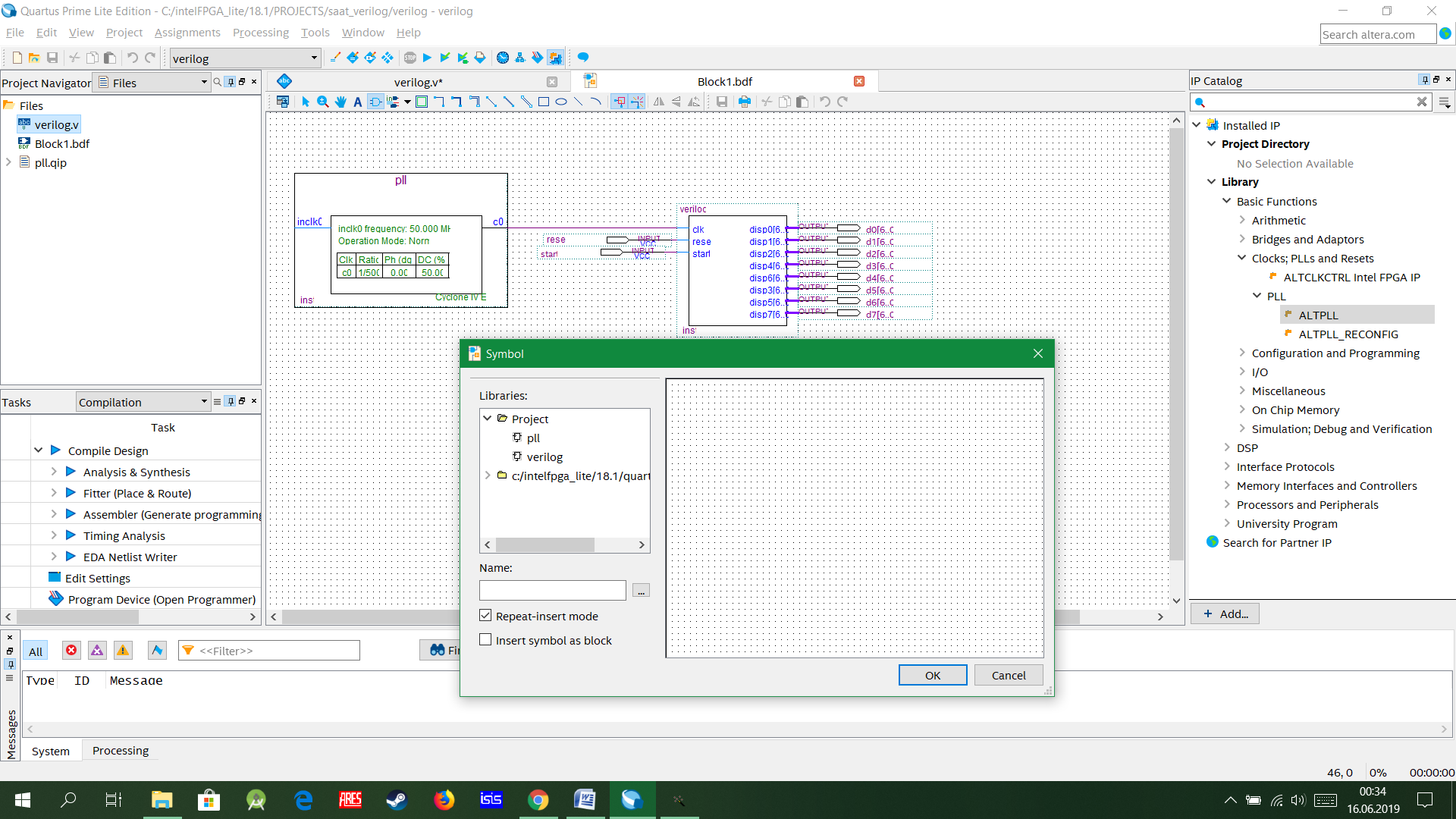


Fig-3 Converting Verilog Code to Block Diagram



**Fig-4** Find that Block Diagram in symbol tools