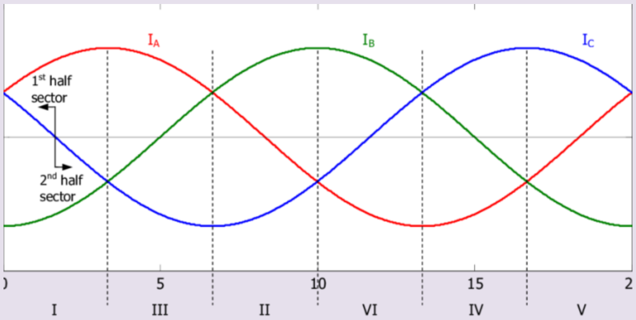
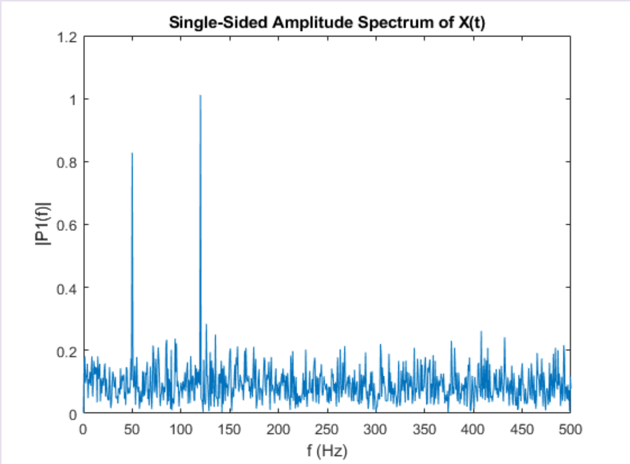
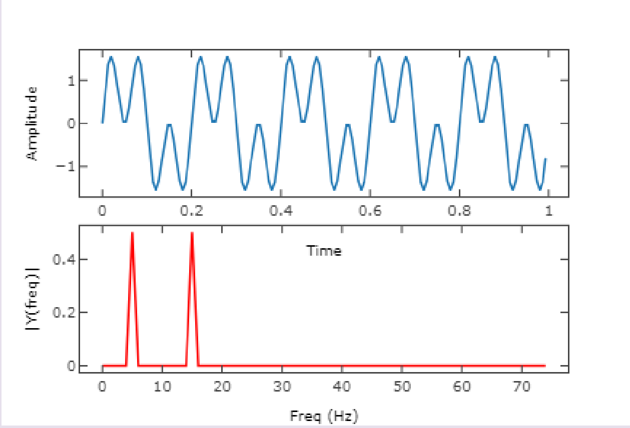
Three Phase Power Analyzer

Developed by Ahmad Al-Astal







CONTENT

CHAPTER ONE:

System Main Features

Tactical Awareness Display (TAD) of A10C Warthog showing night vision of Litening II AT targeting pod inside DCS Simulator searching for possible ground targets 21 Miles away in mid night mission, range is measured using laser range finder fitted with the pod and enabled by the pilot, LSR is the enabling flag, the A10C is close air support (CAS) aircraft designed by Northrop Grumman and it is in operation with United States Air Force USAF.

It is strange that only extraordinary men make the discoveries, which later appear so easy and simple. Georg C. Lichtenberg

Since years reading values from circuit takes lots of effort and it is time consuming operation, for a power circuit with three phase source it is even harder since we have three different voltages and currents flowing through the circuit, if someone wants to study the harmonics of the load currents and voltages it is again a level up of complication.

The main goal of this project is to squeeze analysis time so that studying such systems takes less time and effort, from hours to simply few minutes from taking samples to FFT analysis all is done in just very few steps and the rest is measured, analyzed and stored automatically.

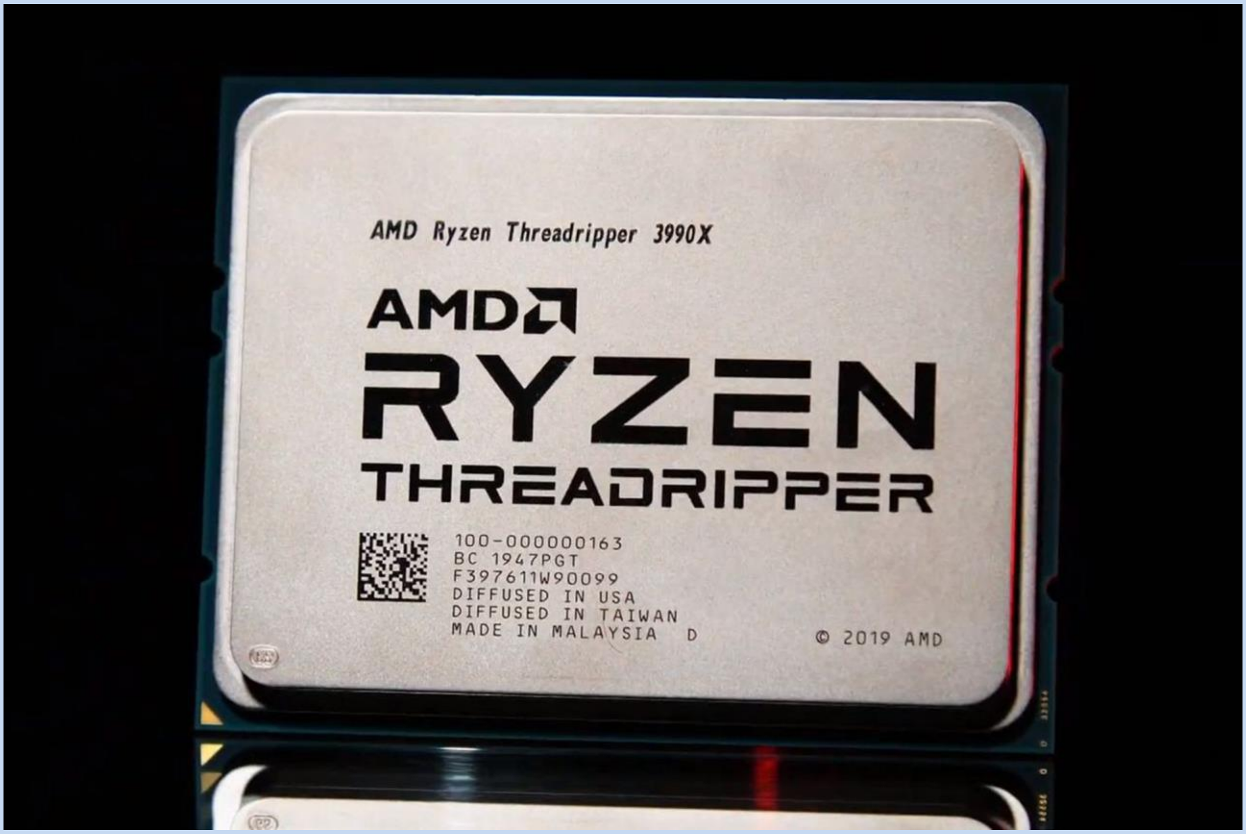
System Main Features

The Three Phase Power Analyzer project scores great features, the main ones are listed below:

- The ability to inject 3^{rd}, 9^{th}, 15^{th} harmonics generated by the FPGA.
- 3 Phase PWM outputs on 6 different pins.
- The 3 Phase injected analog and PWM signals are displayed on VGA screen.
- Reading 3 Phase currents and voltages from load circuit, display them on VGA screen and store them to a host machine, the readings are done through 6 different TLC549 ADCs with 30kHz sampling frequency each.
- NIOS II CPU RISC architecture is implemented inside FPGA Cyclone IV E chip which organizes the storing mechanism to host machine.
- Splendid direct connection between NIOS II and MATLAB.
- MATLAB performs FFT operation according to stored data organized by NIOS II processor.
- MATLAB draws all 3 phase voltages and currents and their FFT in 12-channel oscilloscope.
- Extremely easy to use, with two switches you can control the entire NIOS-MATLAB connection operations.

CHAPTER TWO:

Pin Description



The AMD Ryzen ThreadRipper is a 64-Core CPU with 128 threads, this CPU really costy but that is not for nothing, the Advanced Micro Devices (AMD) crushed all Intel CPUs for the last two years and surprisingly for at least the next year, for now Intel has nothing in hand to face this CPU, this CPU performs every unthinkable operations even for advanced rendering operations and ray trace ones in split second, very powerful one from AMD.

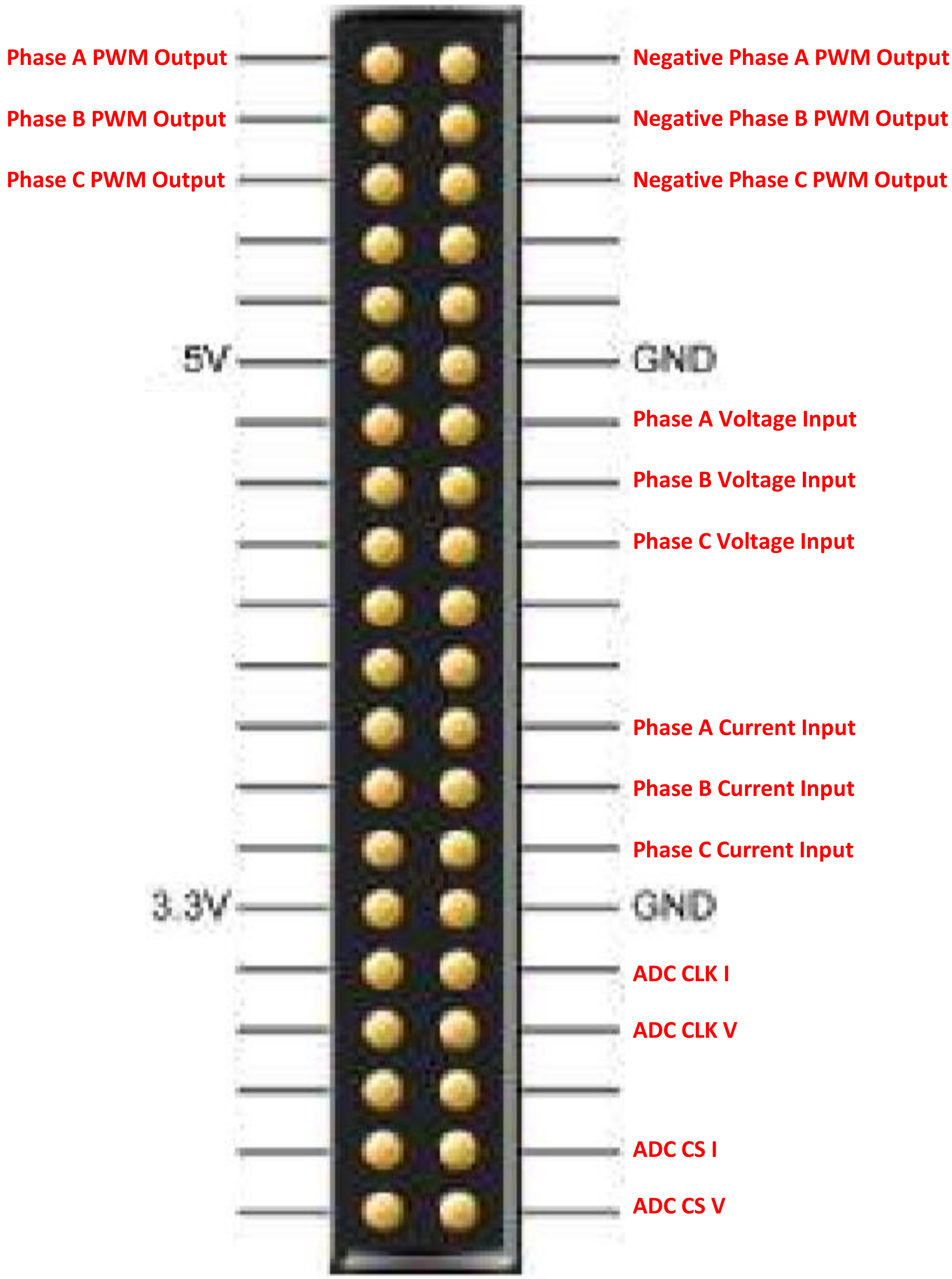
Progress is made by trial and failure, the failures are generally a hundred times more numerous than successes, yet they are usually left unchronicled. William Ramsay

Pin Description

SW#	Description
SW0	+1 Harmonic counter
SW1	+1 Harmonic counter
SW2	+1 Harmonic counter
SW3	Load harmonic
SW4	Start taking sample
SW5	Reset taking sample operation
SW7	Negative 0^{th} harmonic
SW8	Negative 3^{rd} harmonic
SW9	Negative 9^{th} harmonic
SW10	Negative 15^{th} harmonic
SW12	Switch readings between (V) and (A) in page two
SW13	Simulate Jumps activation switch in page one
SW14	+1 Page counter
SW15	+1 Page counter
SW16	+1 Page counter
SW17	Global Reset

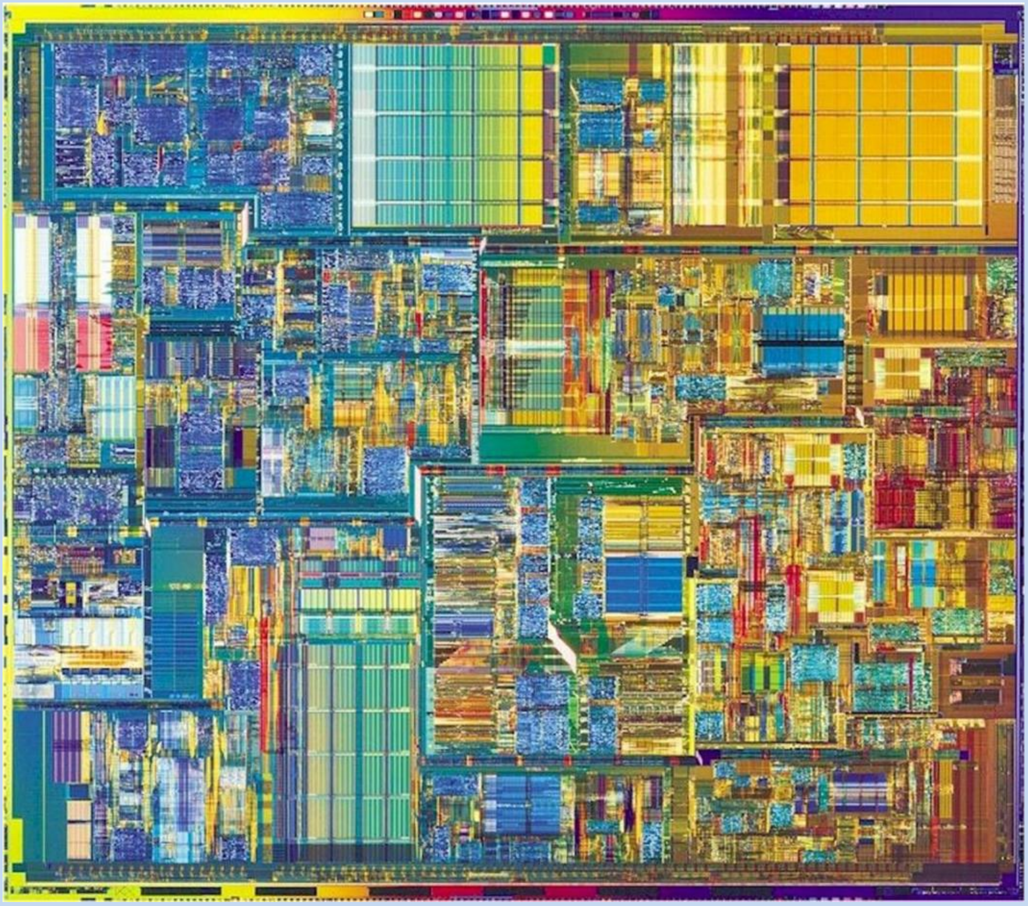
Push#	Description
Push_0	Controls 0^{th} harmonic value
Push_1	Controls 3^{rd} harmonic value
Push_2	Controls 9^{th} harmonic value
Push_3	Controls 15^{th} harmonic value

HEX#	Description
HEX0	0.001 – 0.009
HEX1	0.01 – 0.09
HEX2	0.1 – 0.9
HEX3	1 – 9
HEX4	Chosen harmonic
HEX5	“H” Harmonic indicator

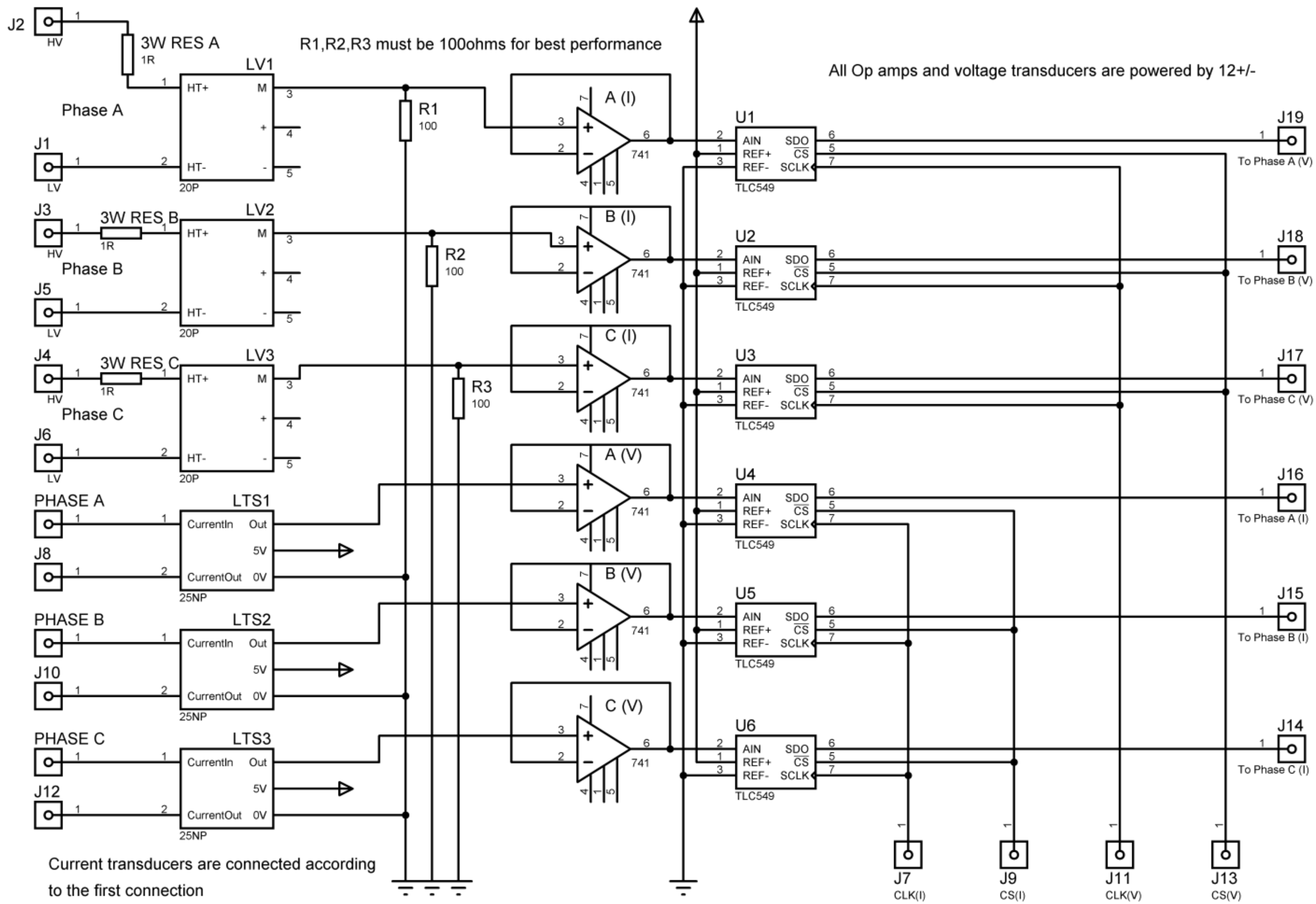

Phase A PWM Output
Negative Phase A PWM Output
Phase B PWM Output
Negative Phase B PWM Output
Phase C PWM Output
Negative Phase C PWM Output
5V
GND
Phase A Voltage Input
Phase B Voltage Input
Phase C Voltage Input
Phase A Current Input
Phase B Current Input
Phase C Current Input
3.3V
GND
ADC CLK I
ADC CLK V
ADC CS I
ADC CS V

CHAPTER THREE:

Circuit Diagram



The Pentium 4 internal CPU architecture, quite old one, this is Intel's last single core CPU before Dual Core and Core2Due CPUs.


R1,R2,R3 must be 100ohms for best performance
All Op amps and voltage transducers are powered by 12+/-
Current transducers are connected according
to the first connection
J2
HV
3W RES A
1R
LV1
HT+
HT-
M
20P
Phase A
J1
LV
J3
HV
3W RES B
1R
LV2
Phase B
J5
LV
J4
HV
3W RES C
1R
LV3
Phase C
J6
LV
PHASE A
J8
LTS1
CurrentIn
Out
5V
CurrentOut
0V
25NP
PHASE B
J10
LTS2
PHASE C
J12
LTS3
R1
100
R2
100
R3
100
A (I)
B (I)
C (I)
A (V)
B (V)
C (V)
741
U1
U2
U3
U4
U5
U6
AIN
REF+
REF-
SDO
CS
SCLK
TLC549
J19
To Phase A (V)
J18
To Phase B (V)
J17
To Phase C (V)
J16
To Phase A (I)
J15
To Phase B (I)
J14
To Phase C (I)
J7
CLK(I)
J9
CS(I)
J11
CLK(V)
J13
CS(V)

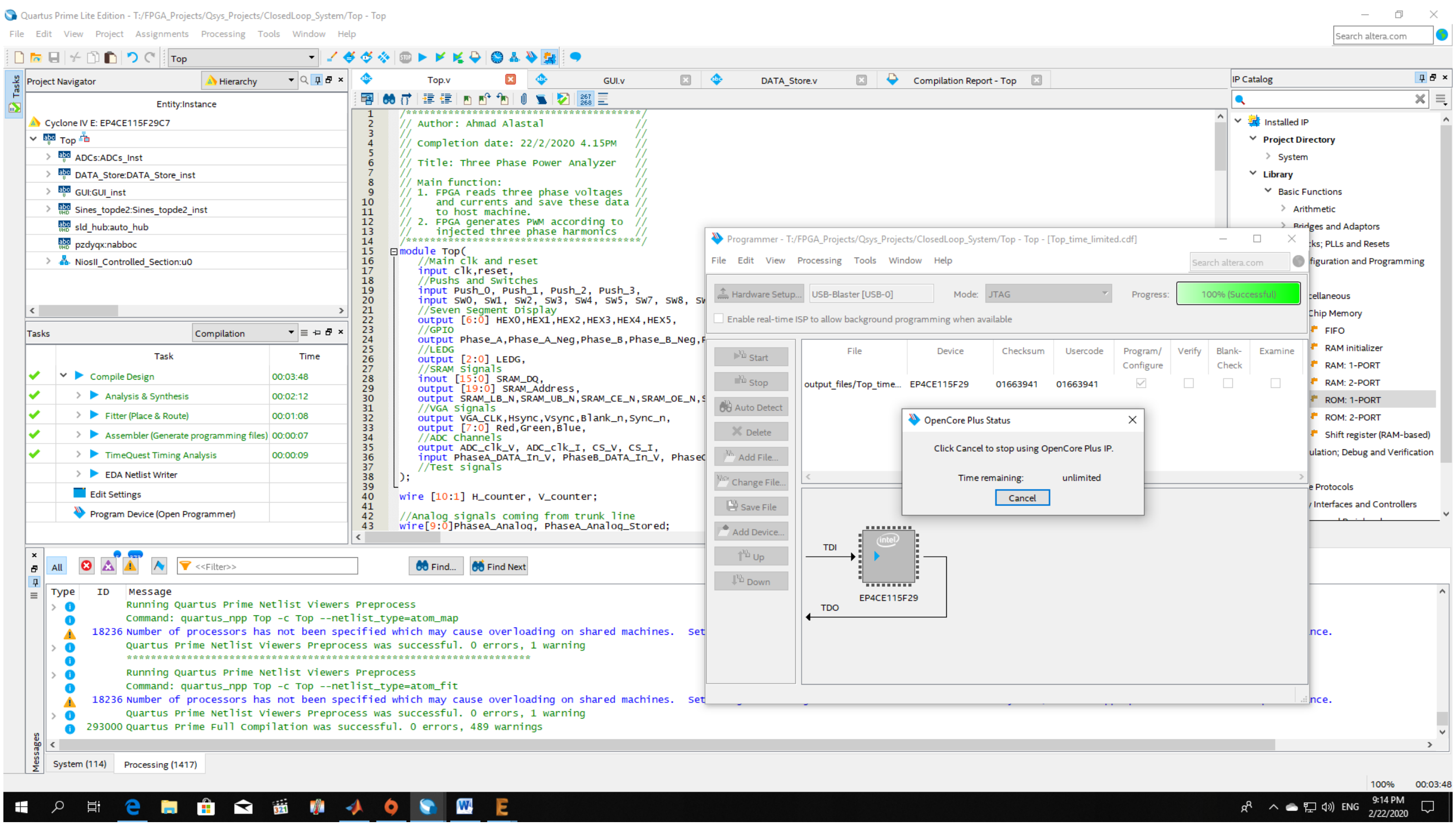
CHAPTER FOUR: Prepare System for Launch

Transporting Proton-M space launch vehicle (SLV) to Baikonur space launch base in Kazakhstan to start launching preparations, the rocket first stage is powered by six RD-275 space rocket engines which produce 1750kNewton of thrust for each at lift off, RD-275s are shown in the picture.

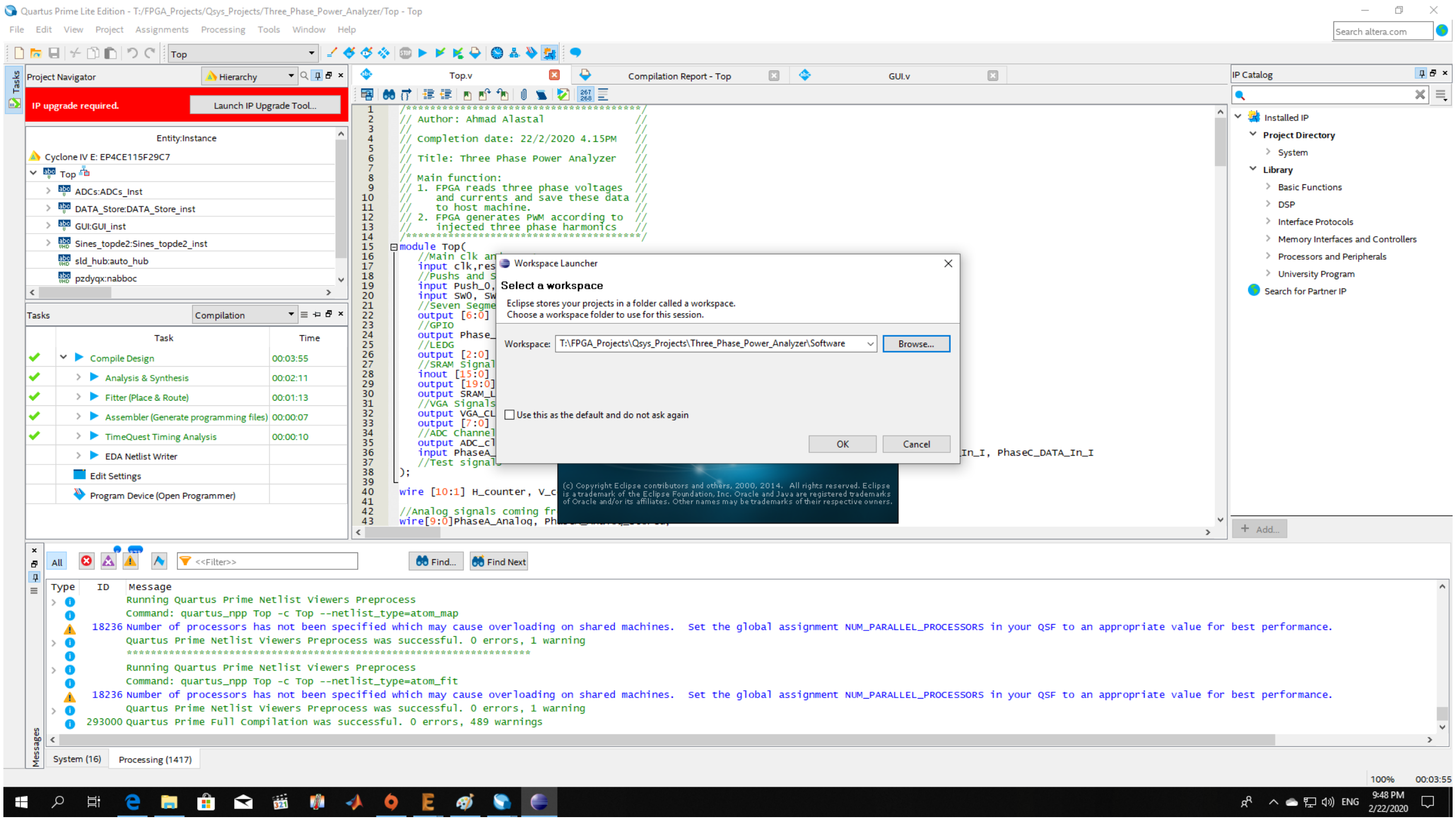
Setup Procedures

Since the system contains NIOS II soft processor we have to do some steps before the system can be fully operational.

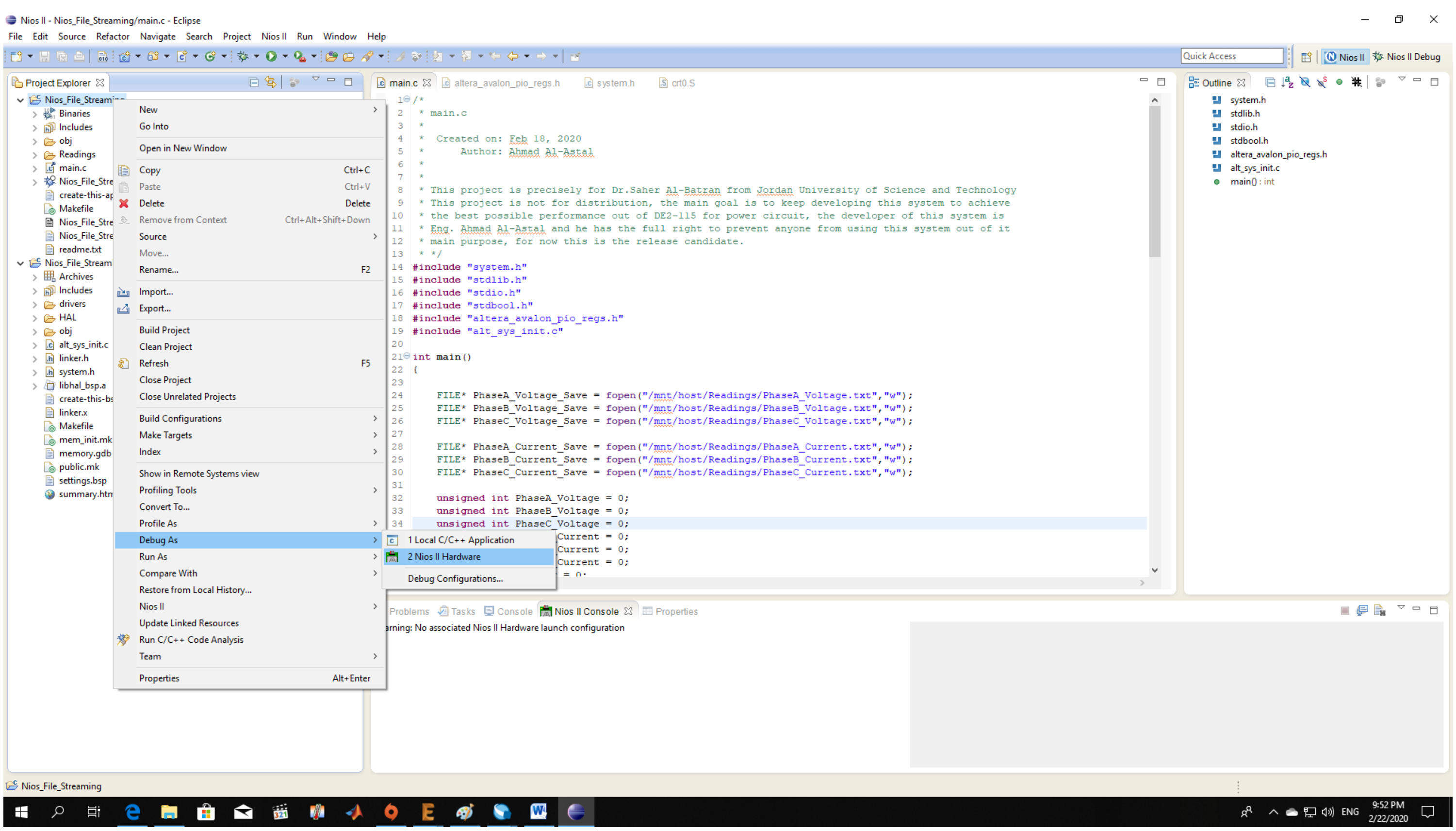
1. Open Quartus 17.0 software and download the .sof file, to download the .sof file go to Tools >> Programmer, make sure the FPGA is connected, from Hardware Setup make sure that USB Blaster is chosen, from Mode make sure that JTAG is chosen, hit Add File and navigate to Top_time_limited.sof which is in ../output_files/Top_time_limited.sof.
   Note: since this is the Lite edition from Quartus, the software adds _time_limited suffix to indicate the free version and limits the NIOS II operation to limited time (1 Hour), it is clear from the below screenshot that a small box pops up with title OpenCore Plus Status, NEVER hit Cancel or the NIOS II will not response to broken JTAG chain.



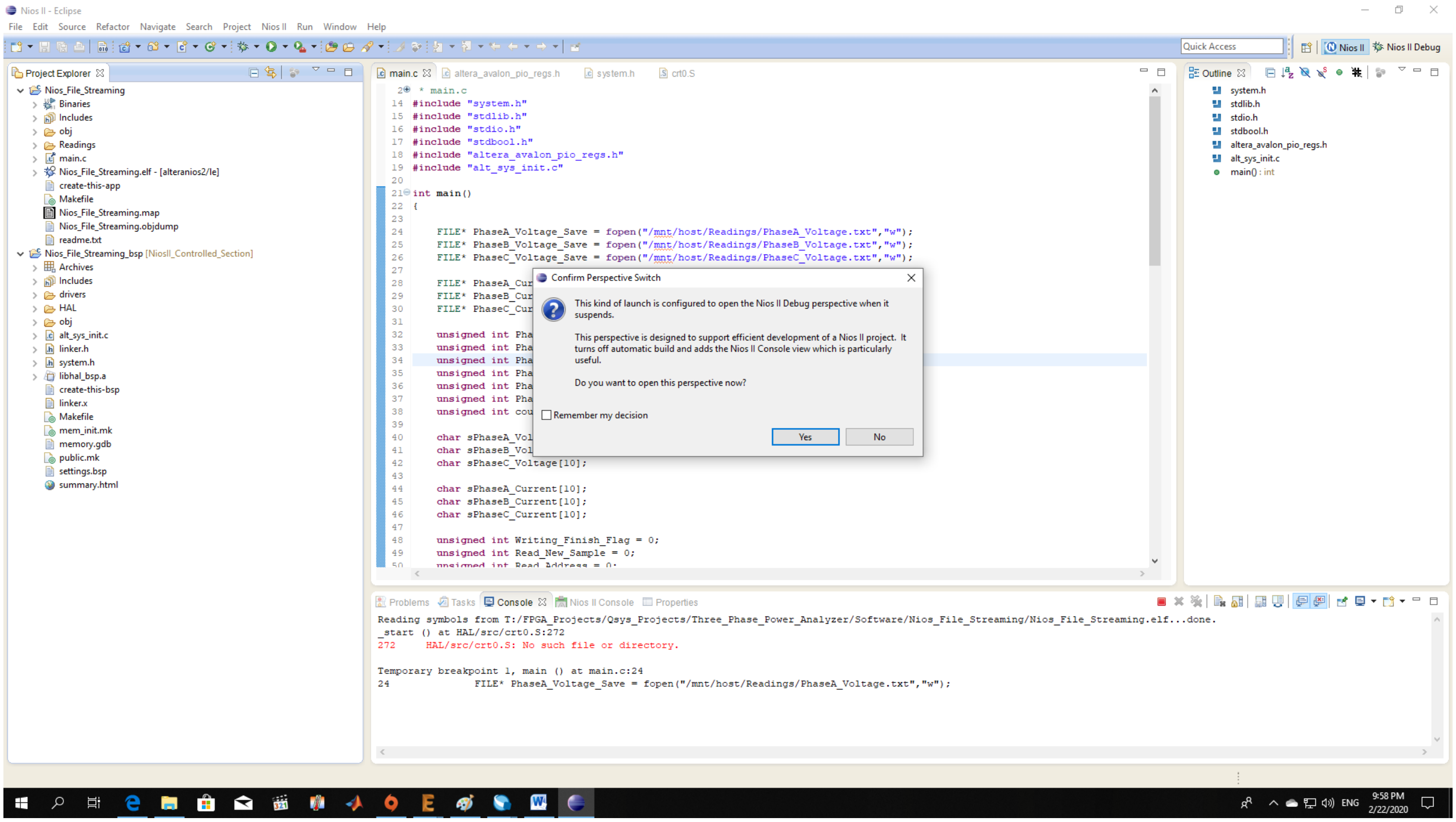
2. Now go to Tools >> Nios II Software Build Tools (SBT) for Eclipse, a small Workspace Launcher box pops up and asks you to navigate to your workspace, navigate to Software folder in the project folder as shown in the next page.



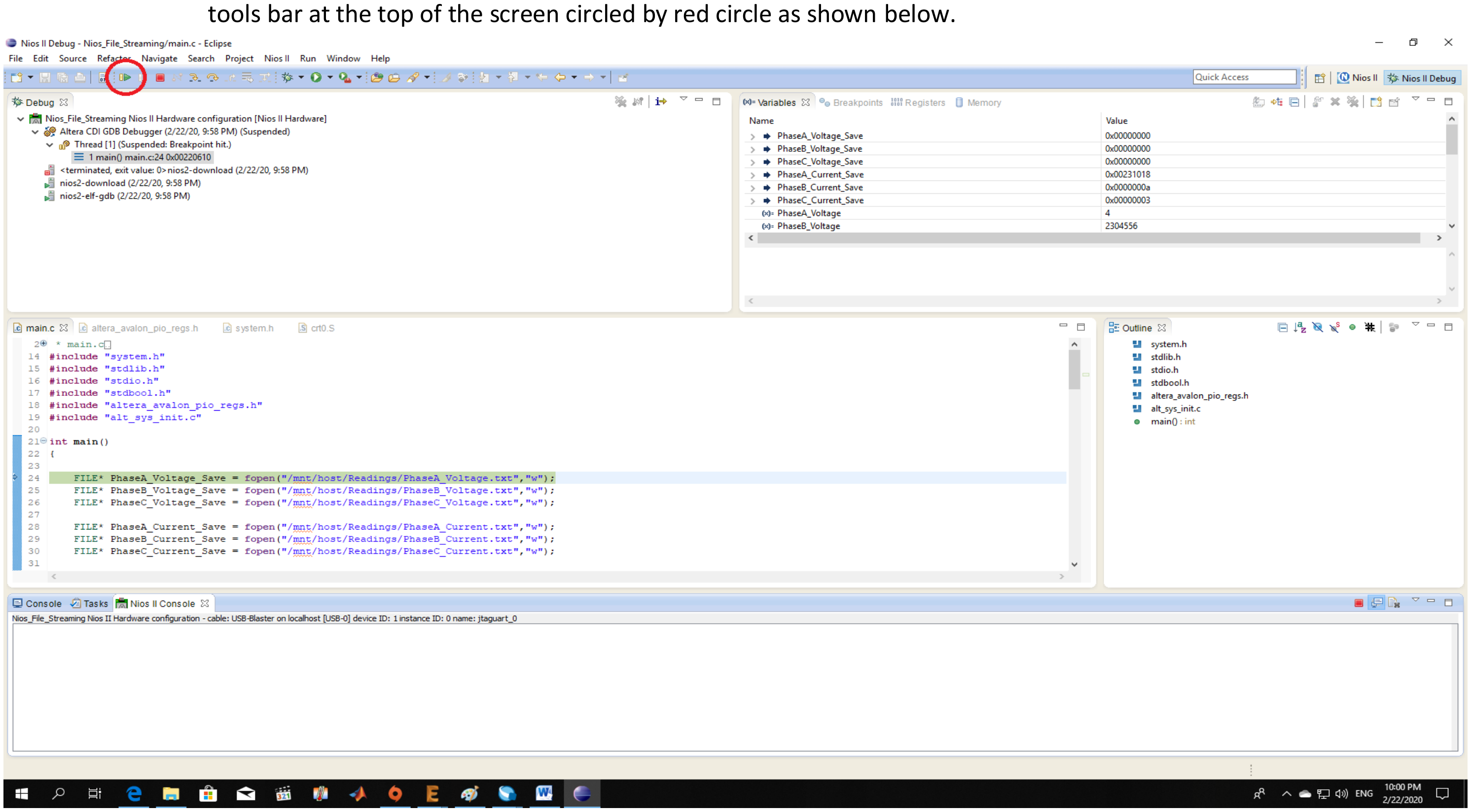
3. To write data to host machine we have to activate the altera_hostfs feature which works only in Debug mode, it is already checked so go to the main project folder >> right click >> Debug As >> Nios II Hardware.



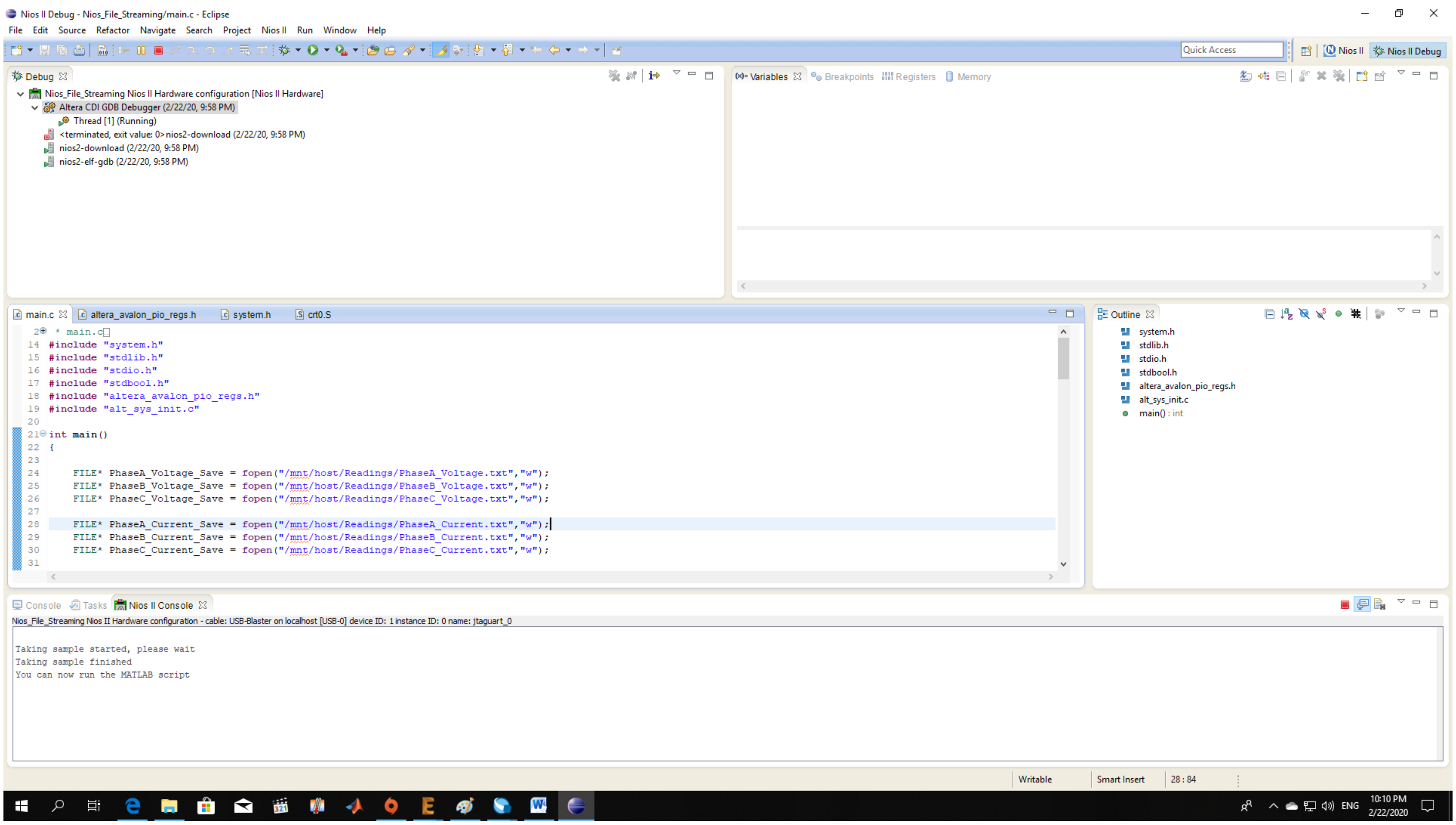
4. It will ask you whether you want to go to the debug screen or not, say yes.



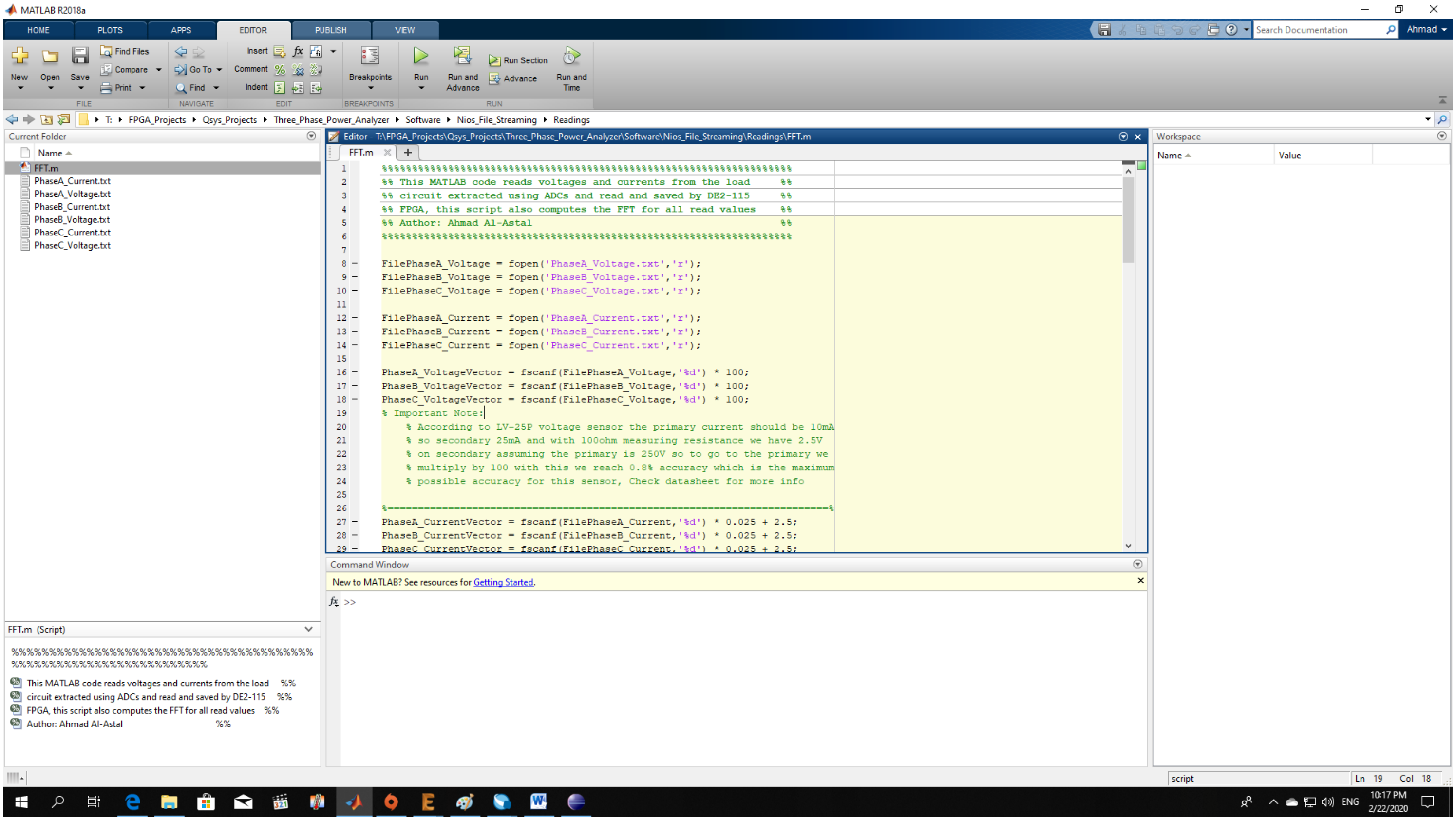
5. You are now in Debug mode screen, you can notice small arrow pointing towards the first executable line of code after the main() function, this arrow indicates that the CPU reached this sentence but still did not execute it yet, we want to keep running the system forever so press F8 on keyboard or Resume in the tools bar at the top of the screen circled by red circle as shown below.



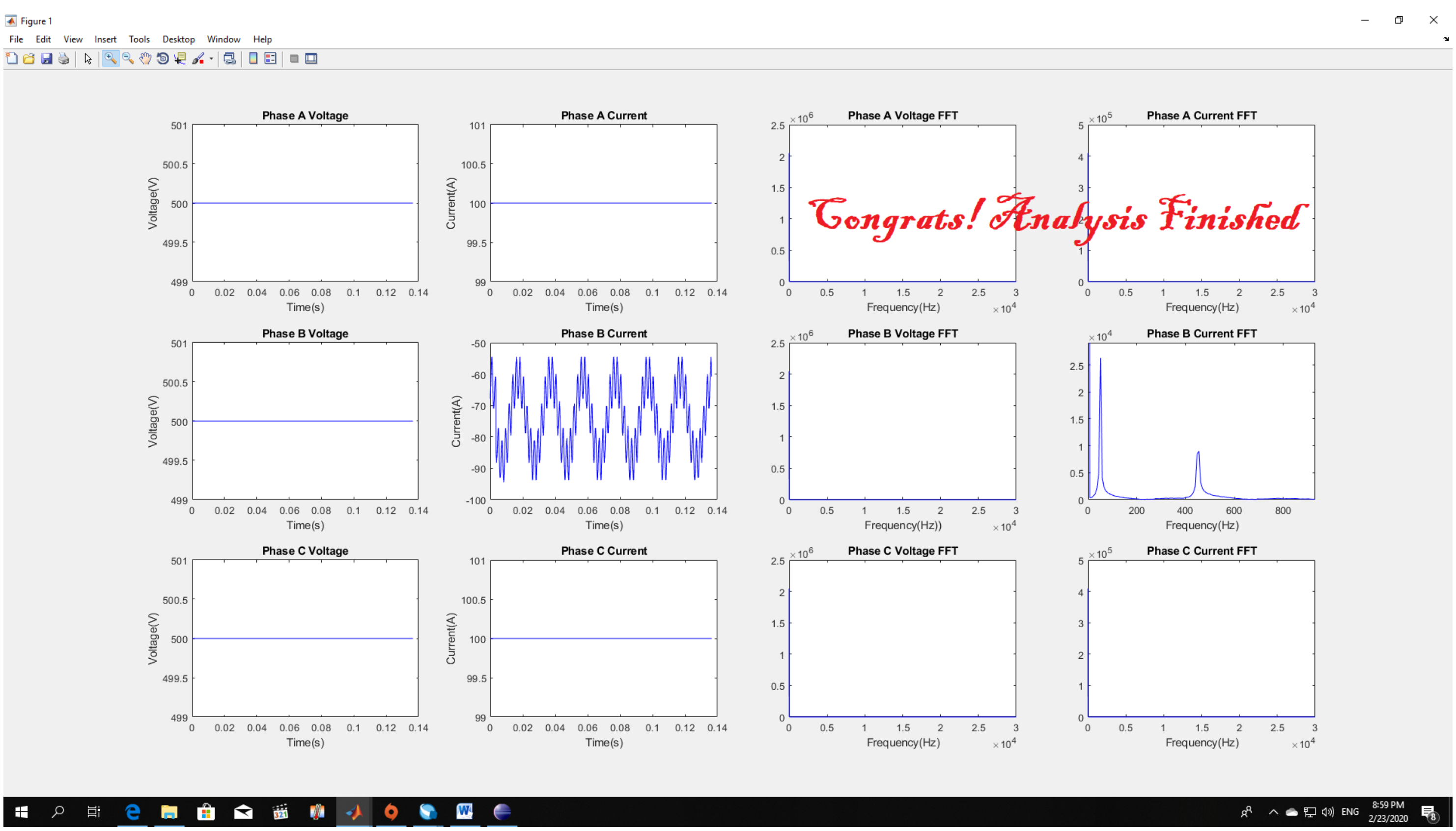
6. You resumed the program but there is still nothing in Nios II Console, now you can control the taking samples operation, to read a sample from load circuit set SW4 to HIGH, you will notice a message appeared in the console, it says "Taking sample started, please wait", wait until the operation finishes, once it is done it will echo to you another message telling you that the operation finished and the data is now ready for analysis, the message says "Taking sample finished, You can now run the MATLAB script", now set SW4 back to LOW, well done the sample is now stored in the Readings folder inside Nios_File_Streaming folder.



7. Now all you have to do is run the MATLAB script, in the Readings folder (../Software/Nios_File_Streaming/Readings) there is a FFT.m MATLAB file, run this file as shown in the next page, if you mistakenly deleted the .txt files adjacent to FFT.m file don't worry the NIOS II processor will generate them for you in the reading sample step so it is safe to manipulate them but not recommended because these values are coming from load circuit and there is a copy of them inside M9K SRAM embedded memory blocks inside FPGA fabric.

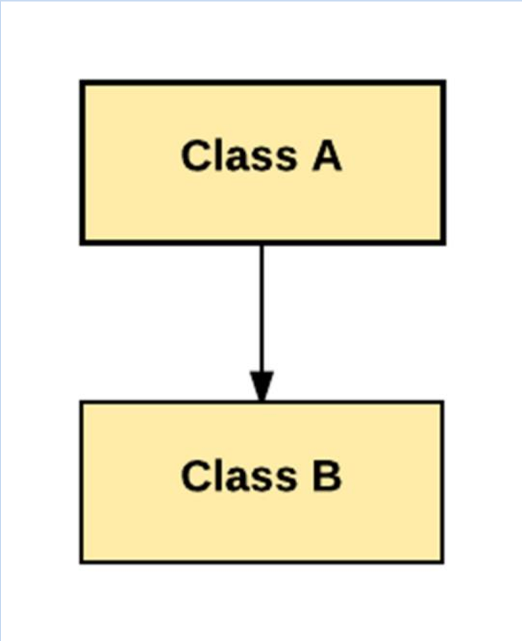


8. For example I am injecting 9th harmonic (450Hz) of sine wave into 50 Hz sine wave system with M = 6.000 and i9 = 0.5 and I am just reading the Phase B current (I don't have 6 ADCs I have only 1 so that output), the result of MATLAB script in shown below, about 20-30 seconds to take 4096 sample points from load circuit, save them to host machine and perform FFT analysis of saved data pretty cool Ha!
Note: You can see once you navigate in FFT diagrams that FFT is symmetrical.



Appendix A:

Inheriting Harmonic Injection SIMULINK Project



Examples

Ex1: Inject the following values to Phase C, $m = 4$, $i_3 = -0.37$, $i_9 = 0.275$, $i_{15} = -0.3$.

Sol:

- Make sure that all switches are set to LOW before begin injecting values.
- Set SW0 and SW1 and SW2 to LOW, now we are at the original signal value(m) and set SW16 and SW17 to HIGH now we activated Phase C control.
- Use push buttons to control HEXs and set the value to 4.000.
- Set Reload SW to HIGH (SW3 to HIGH) then set it back to LOW, the injected value is now valid.
- Set SW0 to HIGH, the harmonic counter is now 1 which means we are at third harmonic (i_3).
- Use push buttons to control HEXs and set the value to 0.3700.
- Since i_3 is negative value set SW8 to HIGH, check tables to know how to control negative values.
- Set SW3 to HIGH then set it back to LOW, the injected value is now valid.
- Set SW1 to HIGH, the harmonic counter is now 2 which mean we are at the ninth harmonic (i_9).
- Use push buttons to control HEXs and set the value to 0.2750.
- Set SW3 to HIGH then set it back to LOW, the injected value is now valid.
- Set SW2 to HIGH, the harmonic counter is now 3 which mean we are at the fifteenth harmonic (i_{15}).
- Use push buttons to control HEXs and set the value to 0.300.
- Since i_{15} is negative value set SW10 to HIGH, check tables to know how to control negative values.
- Set SW3 to HIGH then set it back to LOW, the injected value is now valid.

Ex2: Inject the following values to Phase A, $m = 2.3$, $i_3 = 0.73$, $i_9 = -0.27$, $i_{15} = 0.017$.

Sol:

1. Make sure that all switches are set to LOW before begin injecting values.
2. Set SW0 and SW1 and SW2 to LOW, now we are at the original signal value(m) and set SW16 and SW17 to LOW now we activated Phase A control.
3. Use push buttons to control HEXs and set the value to 2.300.
4. Set Reload SW to HIGH (SW3 to HIGH) then set it back to LOW, the injected value is now valid.
5. Set SW0 to HIGH, the harmonic counter is now 1 which means we are at third harmonic (i_3).
6. Use push buttons to control HEXs and set the value to 0.7300.
7. Set SW3 to HIGH then set it back to LOW, the injected value is now valid.
8. Set SW1 to HIGH, the harmonic counter is now 2 which mean we are at the ninth harmonic (i_9).
9. Use push buttons to control HEXs and set the value to 0.2700.
10. Since i_9 is negative value set SW9 to HIGH, check tables to know how to control negative values.
11. Set SW3 to HIGH then set it back to LOW, the injected value is now valid.
12. Set SW2 to HIGH, the harmonic counter is now 3 which mean we are at the fifteenth harmonic (i_{15}).
13. Use push buttons to control HEXs and set the value to 0.017.
14. Set SW3 to HIGH then set it back to LOW, the injected value is now valid.