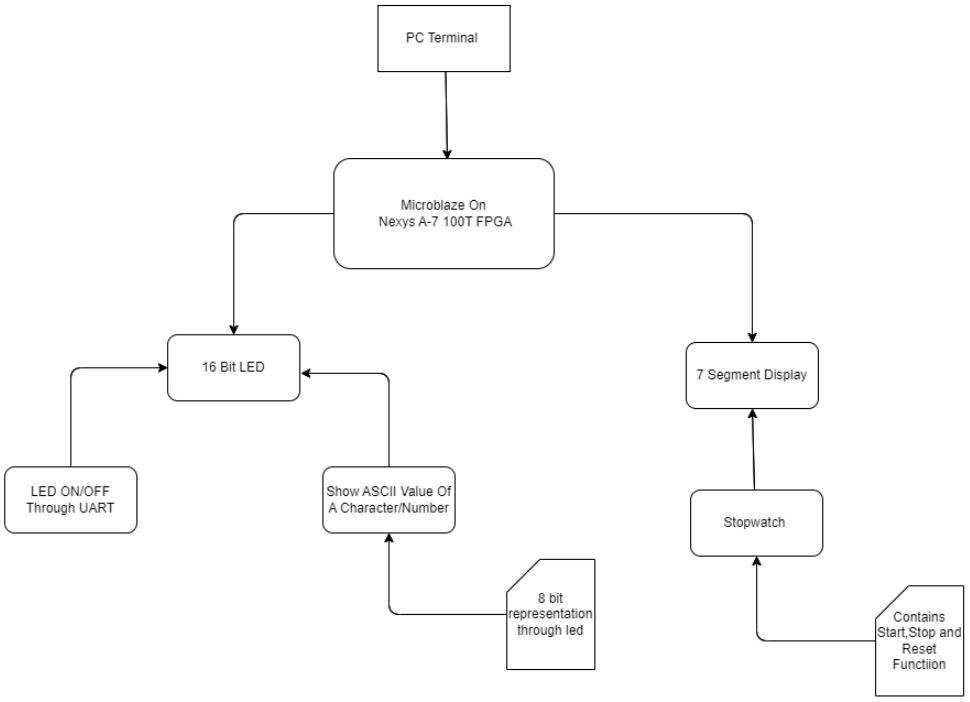
Documentation

13.01.2023

1 Team members

Neaz Mahmud (Mat. Nr. – 2190035)
Manoj Luitel (Mat. Nr. – 2190341)

2 Concept/Idea description

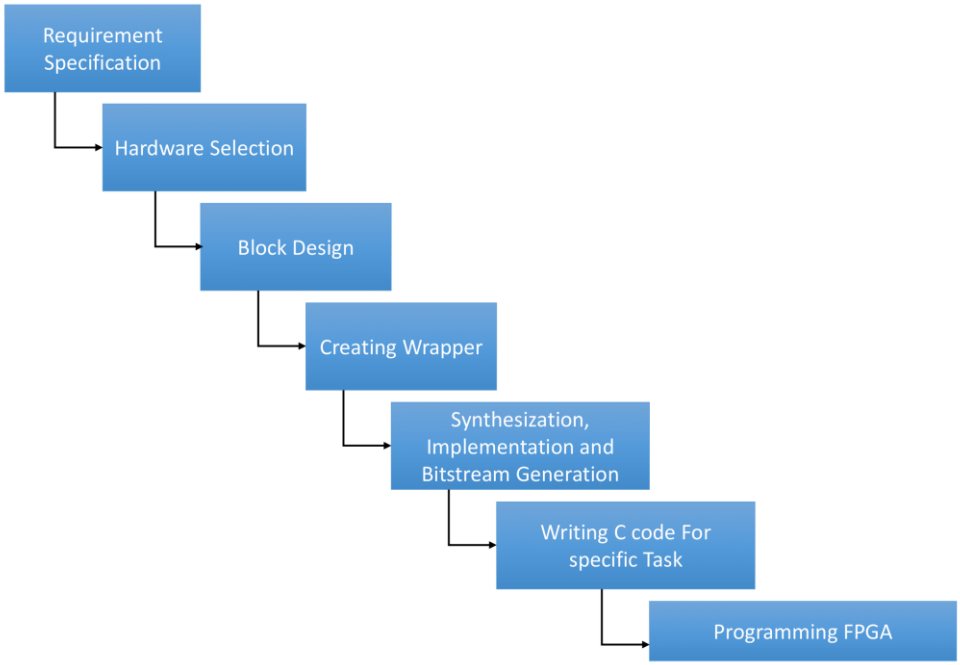


The goal of the project is to implement two independent tasks running parallel in a soft core processor (Microblaze) created in FPGA development board.
The 1st task is the implementation of Stopwatch on 7 segment display FPGA which has Start, Stop and Reset functionality.
The 2nd task is to control (turning on & Off) the on board 16 bit LED through sending command from PC to the FPGA thorough UART Communication.
Furthermore, the 8 bit representation of characters and numbers on LED has also been implemented, Which is also done by UART communication between PC and FPGA board.

3 Project/Team management

3.1 Project Method Used.

The waterfall Model has been used to implement the project work.
A simplified diagram is provided below.



3.2 Task Management

The project consists of two individual tasks as mentioned earlier.

1. Stopwatch implementation
2. Controlling and Showing 8 bit Representation of characters and numbers through UART communication.

Stopwatch implementation was done by **Manoj Luitel.**
UART communication between board and PC to control LED was done by **Neaz Mahmud.**

The documentation was jointly done by **Neaz Mahmud** and **Manoj Luitel.**
The responsible person for the documentation part's name has been provided after every pages in the document.

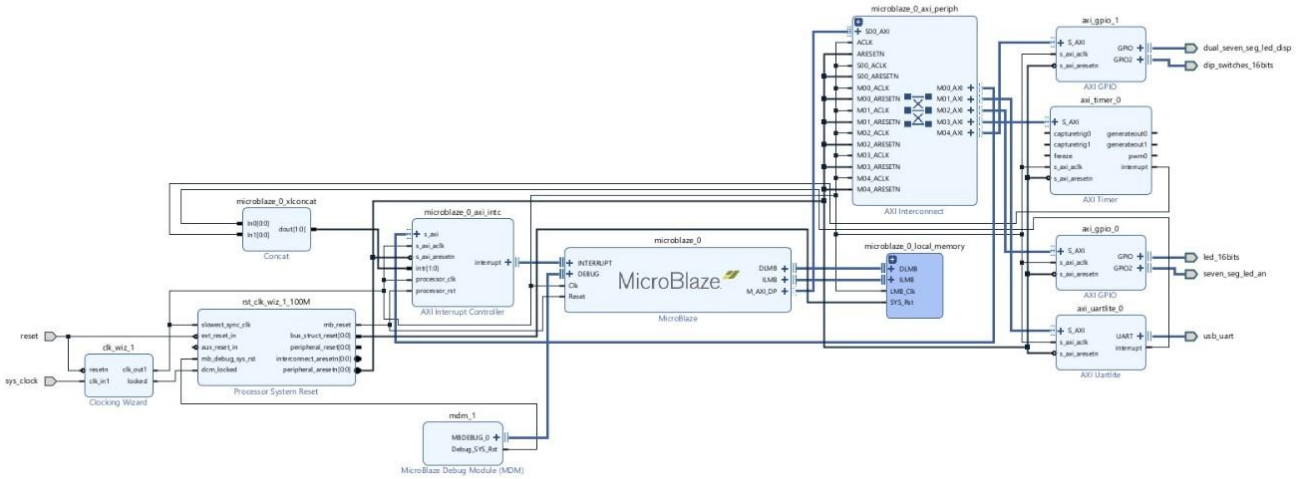
4 Technologies & Hardware Used .

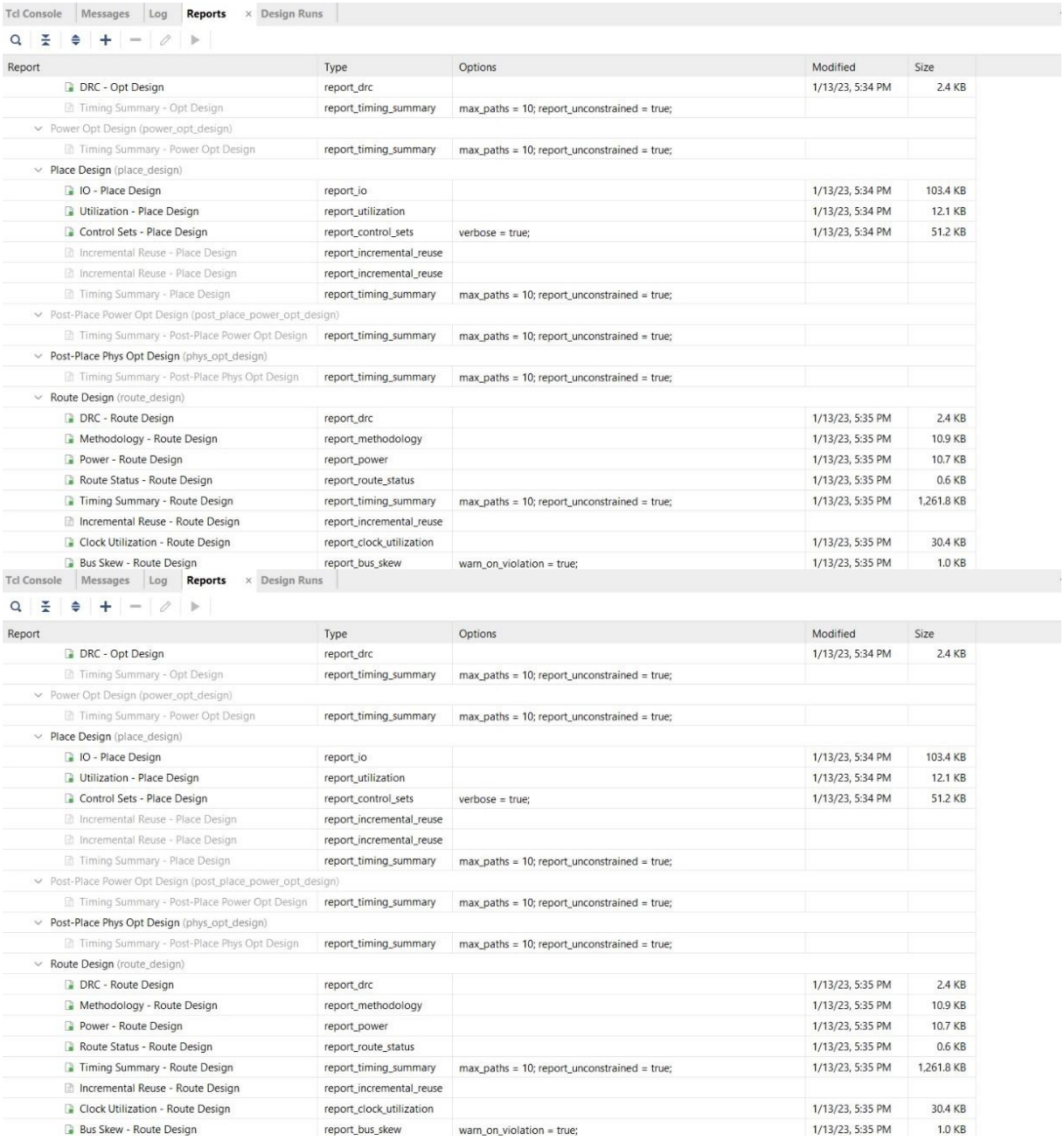
The Project has been Implemented in Nexys A-7 100T development board.
The Nexys A7 100T FPGA is a digital circuit development board that features a Xilinx Artix-7 FPGA. It has 100,000 logic cells, which allows for large and complex designs to be implemented. It also includes on-board memory, USB connectivity, and various digital and analog I/O options. It is commonly used for educational and hobbyist projects, as well as for prototyping and testing in industry.

4.1 For UART Communication part, Verilog has been used as hardware description language.
4.2 For stopwatch part, VHDL has been used as hardware description language.

5 Hardware Implementation

Block design





Tcl Console | Messages | Log | Reports | Design Runs

Report	Type	Options	Modified	Size
DRC - Opt Design	report_drc		1/13/23, 5:34 PM	2.4 KB
Timing Summary - Opt Design	report_timing_summary	max_paths = 10; report_unconstrained = true;		
Power Opt Design (power_opt_design)				
Timing Summary - Power Opt Design	report_timing_summary	max_paths = 10; report_unconstrained = true;		
Place Design (place_design)				
IO - Place Design	report_io		1/13/23, 5:34 PM	103.4 KB
Utilization - Place Design	report_utilization		1/13/23, 5:34 PM	12.1 KB
Control Sets - Place Design	report_control_sets	verbose = true;	1/13/23, 5:34 PM	51.2 KB
Incremental Reuse - Place Design	report_incremental_reuse			
Incremental Reuse - Place Design	report_incremental_reuse			
Timing Summary - Place Design	report_timing_summary	max_paths = 10; report_unconstrained = true;		
Post-Place Power Opt Design (post_place_power_opt_design)				
Timing Summary - Post-Place Power Opt Design	report_timing_summary	max_paths = 10; report_unconstrained = true;		
Post-Place Phys Opt Design (phys_opt_design)				
Timing Summary - Post-Place Phys Opt Design	report_timing_summary	max_paths = 10; report_unconstrained = true;		
Route Design (route_design)				
DRC - Route Design	report_drc		1/13/23, 5:35 PM	2.4 KB
Methodology - Route Design	report_methodology		1/13/23, 5:35 PM	10.9 KB
Power - Route Design	report_power		1/13/23, 5:35 PM	10.7 KB
Route Status - Route Design	report_route_status		1/13/23, 5:35 PM	0.6 KB
Timing Summary - Route Design	report_timing_summary	max_paths = 10; report_unconstrained = true;	1/13/23, 5:35 PM	1,261.8 KB
Incremental Reuse - Route Design	report_incremental_reuse			
Clock Utilization - Route Design	report_clock_utilization		1/13/23, 5:35 PM	30.4 KB
Bus Skew - Route Design	report_bus_skew	warn_on_violation = true;	1/13/23, 5:35 PM	1.0 KB

Tcl Console | Messages | Log | Reports | Design Runs

Report	Type	Options	Modified	Size
DRC - Opt Design	report_drc		1/13/23, 5:34 PM	2.4 KB
Timing Summary - Opt Design	report_timing_summary	max_paths = 10; report_unconstrained = true;		
Power Opt Design (power_opt_design)				
Timing Summary - Power Opt Design	report_timing_summary	max_paths = 10; report_unconstrained = true;		
Place Design (place_design)				
IO - Place Design	report_io		1/13/23, 5:34 PM	103.4 KB
Utilization - Place Design	report_utilization		1/13/23, 5:34 PM	12.1 KB
Control Sets - Place Design	report_control_sets	verbose = true;	1/13/23, 5:34 PM	51.2 KB
Incremental Reuse - Place Design	report_incremental_reuse			
Incremental Reuse - Place Design	report_incremental_reuse			
Timing Summary - Place Design	report_timing_summary	max_paths = 10; report_unconstrained = true;		
Post-Place Power Opt Design (post_place_power_opt_design)				
Timing Summary - Post-Place Power Opt Design	report_timing_summary	max_paths = 10; report_unconstrained = true;		
Post-Place Phys Opt Design (phys_opt_design)				
Timing Summary - Post-Place Phys Opt Design	report_timing_summary	max_paths = 10; report_unconstrained = true;		
Route Design (route_design)				
DRC - Route Design	report_drc		1/13/23, 5:35 PM	2.4 KB
Methodology - Route Design	report_methodology		1/13/23, 5:35 PM	10.9 KB
Power - Route Design	report_power		1/13/23, 5:35 PM	10.7 KB
Route Status - Route Design	report_route_status		1/13/23, 5:35 PM	0.6 KB
Timing Summary - Route Design	report_timing_summary	max_paths = 10; report_unconstrained = true;	1/13/23, 5:35 PM	1,261.8 KB
Incremental Reuse - Route Design	report_incremental_reuse			
Clock Utilization - Route Design	report_clock_utilization		1/13/23, 5:35 PM	30.4 KB
Bus Skew - Route Design	report_bus_skew	warn_on_violation = true;	1/13/23, 5:35 PM	1.0 KB

Synthesis Report

5.1 Microblaze Soft Processor Core

The MicroBlaze CPU is highly, flexible preset 32-bit/64-bit RISC microprocessor configurations. System designers can leverage the Vitis core development kit in 2019.2, or the Eclipse-based Xilinx Software Development Kit (SDK) in 2019.1 or earlier to start developing for the MicroBlaze processor using select evaluation kits, with no prior FPGA experience. The MicroBlaze processor meets the requirements of many diverse applications including Industrial, Medical, Automotive, Consumer, and Communications markets. FPGA designers use the Vivado HL/ML Edition design tools to setup the MicroBlaze processor to any supported Xilinx device. Starting from one of these configurable presets, more customization is possible from a variety of specific processor options and a catalog of driver-

enabled drag and drop peripherals such as PWMs, UARTs, DMAs, serial interfaces, to satisfy the specific needs of the application. It is also available as part of legacy IDS embedded edition for older FPGA device families like Spartan®-6.

5.2 Clocking Wizard

Vivado's Clocking Wizard is responsible to configure a CMT to produce any required clock signals. The Wizard lets you enter your desired clock frequencies and select a few signal properties, and then it produces a Verilog/VHDL module that you can include in your design.

5.3 AXI Interrupt Controller

The LogiCORE IP AXI Interrupt Controller (AXI INTC) concentrates multiple interrupt inputs from peripheral devices to a single interrupt output to the system processor. This AXI INTC core is designed to interface with the AXI4-Lite protocol.

Microblaze Debug Module
MDM is the configurable software access module having debug functionality through the AXI4-Lite interface It supports for cross-trigger between connected MicroBlaze cores, Zynq-7000 Processing System, and Integrated Logic Analyzer (ILA) cores. It has external trace function to funnel program trace from connected MicroBlaze cores to external interfaces.

5.4 AXI GPIO

The AXI GPIO provides a general purpose input/output interface to the AXI (Advanced extensible Interface) interface. This 32-bit soft IP core is designed to interface with the AXI4-Lite interface. This document contains information about the AXI4 version of the core.

5.5 AXI Timer

It is a 32-bit timer module that attaches to the AXI4-Lite interface. In Cascade mode, it can be used as 64-bit timer module. It has two programmable interval timers with interrupt, event generation, and event capture capabilities.

5.6 AXI Interconnect

The AXI Interconnect IP connects one or more AXI memory-mapped Master devices to one or more memory-mapped Slave devices.
The AXI interfaces conform to the AMBA AXI version 4 specifications from ARM, including the AXI4-Lite control register interface subset.

6 Software Implementation

C language has been used to implement the code.

7 Sources/References

https://support.xilinx.com/s/?language=en_US
https://digilent.com/reference/programmable-logic/nexys-a7/reference-manual
https://support.xilinx.com/s/question/0D52E000074KqYaSAK/microblaze-hello-world-over-uart?language=en_US
https://www.xilinx.com/support.html#documentation
https://digilent.com/reference/learn/programmable-logic/tutorials/arty-getting-started-with-microblaze/start