Marwadi University	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology
Subject: Capstone Project (01CT1718)	System Design and Architecture
	Date: 19/09/2025 | Enrolment No: 92200133031

System Design and Architecture

Introduction:

The objective of the Capstone project is to build reliable communication protocols like UART/I2C over FPGA board on Altera Cyclone II or Nexys4 DDR to easily interface sensors and different module to interface the Input and Output via FPGA boards using the unique features of the FPGA board. It will help the people to understand the FPGA and communication protocol structure and use of it while testing.

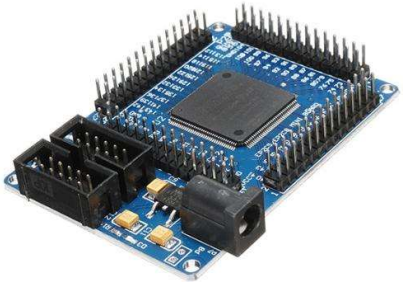
Project Proposal:

A Field-Programmable Gate Array (FPGA) is a semiconductor device based on a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. FPGAs are reconfigurable, meaning that the logic functions within the FPGA can be reprogrammed to implement different designs or algorithms after manufacturing. This reconfigurability contrasts with fixed-function Application-Specific Integrated Circuits (ASICs), making FPGAs highly versatile for prototyping and deployment in a wide range of applications.

The FPGA boards are quite faster than usual microcontroller because of its capability to adapt the advantage of different parameter like high speed parallel processing, custom GPIO interfaces, frequency trading, While FPGAs offer superior hardware-level performance and flexibility, microcontrollers are generally better for cost-sensitive or low-power applications with sequential tasks, thanks to their easier development process and lower unit cost.

Here we can use FPGA board of Altera Cyclone II or Xilinx Nexys 4 DDR, which are very well known FPGA Board for implementing Prototype of HDL design on it and test its compatibility for moving forward the Chip manufacturing or for making a design for an system in less time to build application with low cost.

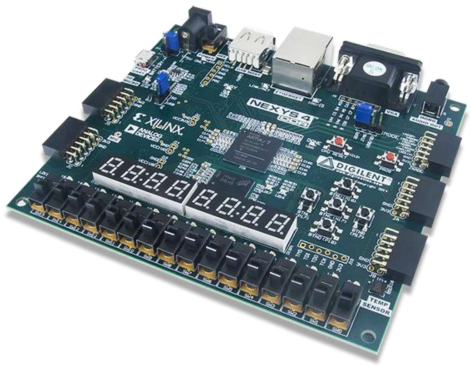
Altera Cyclone II with EP2C5T144 chip



- EPROM chip configuration using EPCS4, the size of 4Mbit
- Onboard 50M active patch crystal (Crystal Slaughter halfback)
- Power supply with a largemouth outlet, single 5V power supply
- Board has power indicator and resets switch
- Onboard 3 SMD LED, the LED test can be done an experiment, more experiments using lead to complete

Marwadi University	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Capstone Project (01CT1718)	System Design and Architecture	
	Date: 19/09/2025	Enrolment No: 92200133031

Nexys 4 DDR with XC7A100TCS0324A



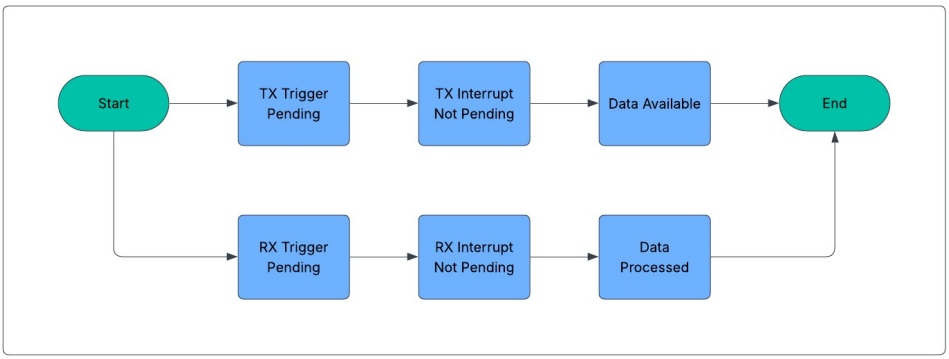
Nexys 4 DDR

Artix-7 FPGA

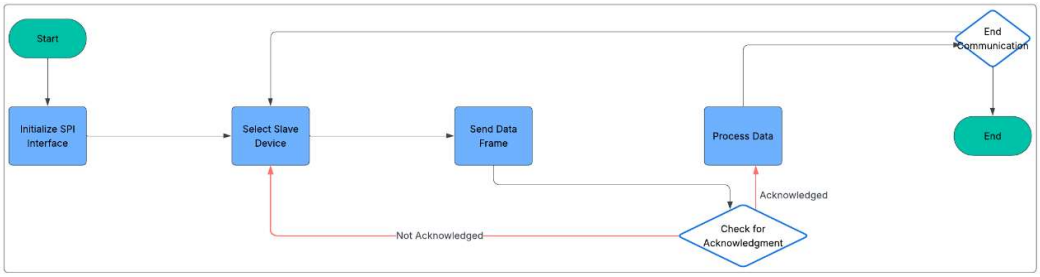
Features	
• Programmable over JTAG and Quad-SPI Flash • On-chip analog-to-digital converter	
Key Specifications	
FPGA Part #	XC7A100T-1CSG324C
Logic Slices	15,850 (4 6-input LUTs & 8 flip-flops each)
Block RAM	4,860 Kbits
Clock Tiles	6 (each with PLL)
DSP Slices	240
Internal clock	450 MHz+
DDR2	128 MiB
Cellular RAM	16MB
Ethernet	10/100 PHY
Connectivity and Onboard I/O	
SD	microSD card connector
Pmod Connectors	4 Pmod ports
VGA	12-bit VGA port
Audio	PWM audio output
Microphone	PDM mic
Temp sensor	One temperature sensor
Display	2 4-digit seven segment displays
Switches	16
Buttons	4
LEDs	16
Tri-color LEDs	2

Flow Diagram for each communication protocols:

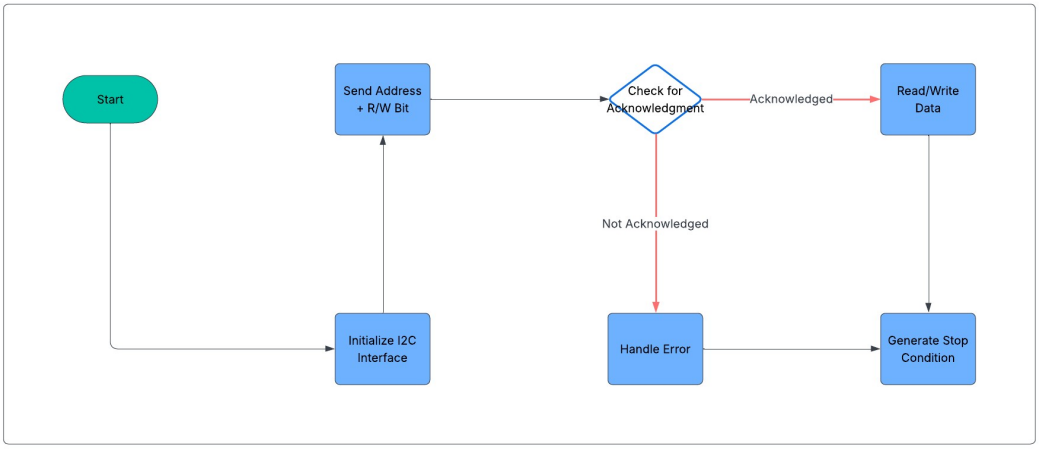
UART



SPI



I2C



Modular Design:

The system is divided into independent modules to simplify development and testing. Each communication protocol is written as a separate Verilog module for reusability.

The below diagram is of UART module where we can see the 3 different modules inside it Baud Rate generator, Transmitter and Receiver. Each module designed such way in Verilog that can controlled via one single top module to connect them together.

The Baud rate, expressed in bits per second (bps) is the rate of symbols transferred across a transmission medium per second. The standard baud rates are the following: 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600.

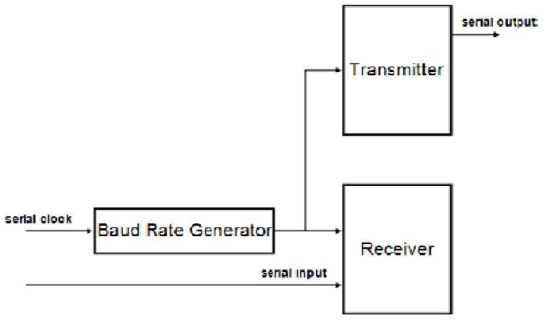


Fig. 1 UART Module

The Verilog module can create a pulse signal at a much slower rate than the main system clock, so it can be used for UART communication (serial transmission at a given baud rate like 9600, 115200, etc.).

The FPGA clock is much faster than the baud rate.

For example:

- Clock = 100,000,000 Hz (100 MHz)
- Baud rate = 9600 bits per second,

To match 9600 baud, the code figures out how many 100 MHz cycles equal 1 baud tick.

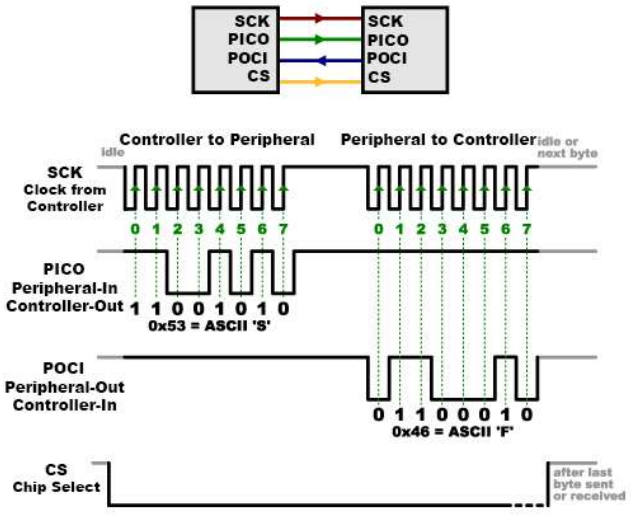
Formula used:

- **COUNTER_LIMIT = (CLOCK_FREQ + BAUD_RATE/2) / BAUD_RATE - 1**
- System Clock ÷ Baud Rate = Number of cycles per bit (rounded properly).

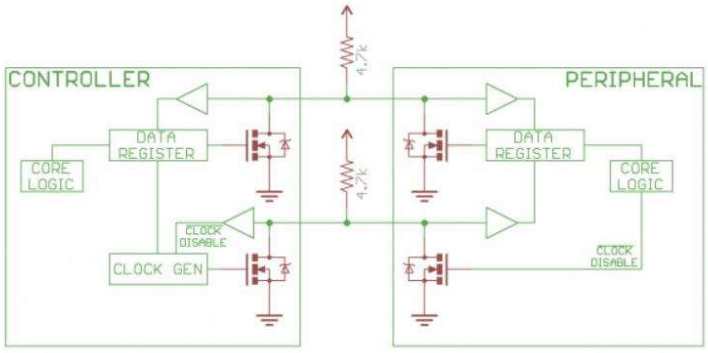
How it will work:

1. A counter starts at 0 and increases every clock cycle.
2. When it reaches COUNTER_LIMIT, that means one baud interval has passed.
3. At this moment
4. The counter resets to 0.
5. **baud_clk_en** goes HIGH for one cycle (a pulse).
6. Otherwise, the counter just keeps counting and **baud_clk_en** stays low.

The image shown below is for SPI modular design, where we can add multiple peripheral/slaves which can be controlled by master

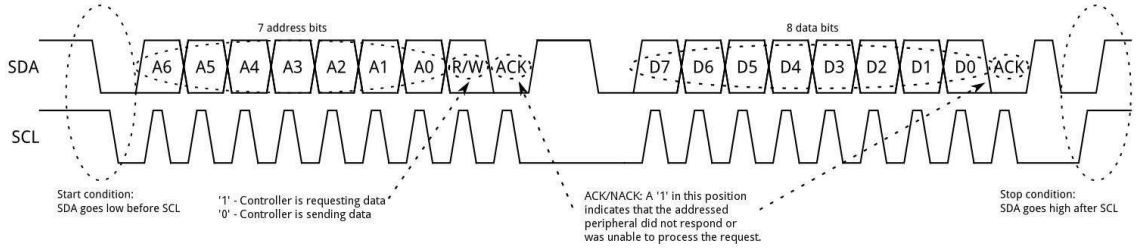


In the last the given below is modular/architecture hardware design of the I2C protocol which we can follow



Notice the two pull-up resistors on the two communication lines.

The I2C packet framework:



While exploring the FGPA datasheet we get to know that the Nexys 4DDR have inbuilt peripherals or say sensors which are capable of doing I2C and SPI communication like for SPI protocol there is 3 Axis Accelerometer which detects acceleration towards different 3 axis XYZ and give 16 bit data from it, the sensor is originally made from Analog Devices named as adxl362.

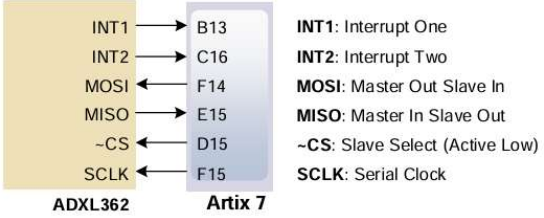


Figure 23. Accelerometer interface.

The resister map shows what kind of data provided that

REGISTER MAP

Table 11. Register Summary

Reg	Name	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset	RW
0x00	DEVID_AD	[7:0]	DEVID_AD[7:0]								0xAD	R
0x01	DEVID_MST	[7:0]	DEVID_MST[7:0]								0x1D	R
0x02	PARTID	[7:0]	PARTID[7:0]								0xF2	R
0x03	REVID	[7:0]	REVID[7:0]								0x01	R
0x08	XDATA	[7:0]	XDATA[7:0]								0x00	R
0x09	YDATA	[7:0]	YDATA[7:0]								0x00	R
0x0A	ZDATA	[7:0]	ZDATA[7:0]								0x00	R
0x0B	STATUS	[7:0]	ERR_USER_REGS	AWAKE	INACT	ACT	FIFO_OVER-RUN	FIFO_WATER-MARK	FIFO_READY	DATA_READY	0x40	R
0x0C	FIFO_ENTRIES_L	[7:0]	FIFO_ENTRIES_L[7:0]								0x00	R
0x0D	FIFO_ENTRIES_H	[7:0]	UNUSED						FIFO_ENTRIES_H[1:0]		0x00	R
0x0E	XDATA_L	[7:0]	XDATA_L[7:0]								0x00	R
0x0F	XDATA_H	[7:0]	SX				XDATA_H[3:0]				0x00	R
0x10	YDATA_L	[7:0]	YDATA_L[7:0]								0x00	R
0x11	YDATA_H	[7:0]	SX				YDATA_H[3:0]				0x00	R
0x12	ZDATA_L	[7:0]	ZDATA_L[7:0]								0x00	R
0x13	ZDATA_H	[7:0]	SX				ZDATA_H[3:0]				0x00	R
0x14	TEMP_L	[7:0]	TEMP_L[7:0]								0x00	R
0x15	TEMP_H	[7:0]	SX				TEMP_H[3:0]				0x00	R
0x16	Reserved	[7:0]	Reserved[7:0]								0x00	R

From the different resister we can use read and write command by creating Master module in form of Verilog code such that FPGA board can communicate with the sensor,

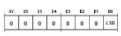
These are data format that we can get from the sensor

X-AXIS DATA REGISTERS

These two registers contain the sign extended (SX) x-axis acceleration data. XDATA_L contains the eight least significant bits (LSBs), and XDATA_H contains the four most significant bits (MSBs) of the 12-bit value.

The sign extension bits (B[15:12], denoted as SX in the XDATA_H bit map that follows) have the same value as the MSB (B11).

Address: 0x0E, Reset: 0x00, Name: XDATA_L



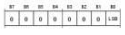
Address: 0x0F, Reset: 0x00, Name: XDATA_H

Y-AXIS DATA REGISTERS

These two registers contain the sign extended (SX) y-axis acceleration data. YDATA_L contains the eight LSBs and YDATA_H contains the four MSBs of the 12-bit value.

The sign extension bits (B[15:12], denoted as SX in the YDATA_H bit map that follows) have the same value as the MSB (B11).

Address: 0x10, Reset: 0x00, Name: YDATA_L



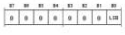
Address: 0x11, Reset: 0x00, Name: YDATA_H

Z-AXIS DATA REGISTERS

These two registers contain the sign extended (SX) z-axis acceleration data. ZDATA_L contains the eight LSBs, and ZDATA_H contains the four MSBs of the 12-bit value.

The sign extension bits (B[15:12], denoted as SX in the ZDATA_H bit map that follows) have the same value as the MSB (B11).

Address: 0x12, Reset: 0x00, Name: ZDATA_L

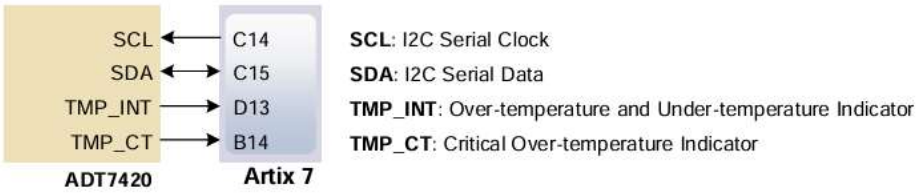


Address: 0x13, Reset: 0x00, Name: ZDATA_H

There are 16 bits of data are there for each axis acceleration, here we can map our data from MSB and take it form of LED like 16/3 = 5. Each axis can show 5 bit data.

The MSB axis data contains 4 bit sign bit so we can avoid that to get bit 11 to bit 7 data, which can be later converted to seven segment to display it on digits too.

The Nexys 4 DDR also contains I2C peripheral/sensor of temperature sensor on the right corner which is internally connected to main IC with I2C parameter. The sensor is also provided on the FPGA board by Analog Devices which is named as ADT7420.



From the datasheet we can see the structure of the sensor,

SERIAL INTERFACE

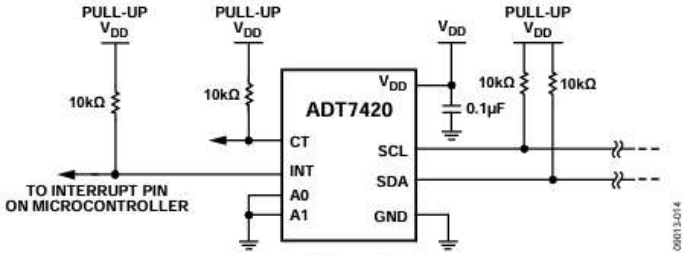


Figure 13. Typical I²C Interface Connection

the bus address of the sensors can be configured via multiple options, but here we will choose 0x4B in out Verilog HDL code to interface the temperature data

SERIAL BUS ADDRESS

Like most I²C-compatible devices, the ADT7420 has a 7-bit serial address. The five MSBs of this address for the ADT7420 are hardwired internally to 10010. Pin A1 and Pin A0 set the two LSBs. These pins can be configured two ways, low and high, to give four different address options. Table 20 shows the different bus address options available. The recommended pull-up resistor value on the SDA and SCL lines is 10 kΩ.

Table 20. I²C Bus Address Options

Binary							
A6	A5	A4	A3	A2	A1	A0	Hex
1	0	0	1	0	0	0	0x48
1	0	0	1	0	0	1	0x49
1	0	0	1	0	1	0	0x4A
1	0	0	1	0	1	1	0x4B

In other parameter we can follow the bit format

Read

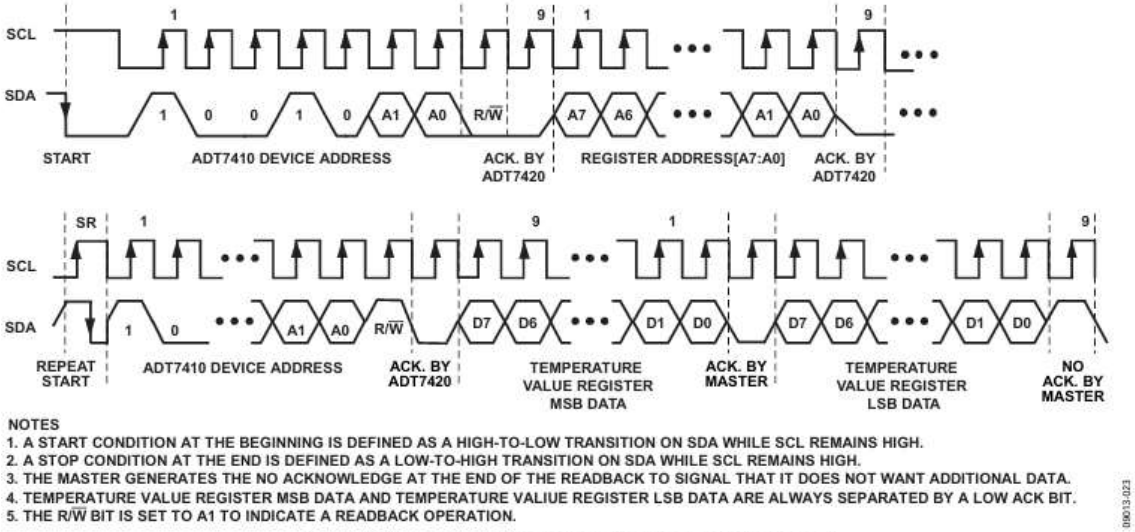


Figure 17. Reading Back Data from the Temperature Value Register

Technology Stack:

Hardware:

- FPGA Boards: Xilinx Nexys 4 DDR
- Sensors/Peripherals: I2C (temperature), UART(self,), SPI (accelerometer).

Software/Tools:

- HDL Language: Verilog HDL – synthesizable, industry standard.
- Design & Synthesis Tools:
  - Quartus II (for Altera).
  - Vivado (for Xilinx).
- Simulation: ModelSim – to test waveform and behavior.

Verilog is widely used for FPGA and ASIC design, making it industry relevant. Quartus II and Vivado are free tools for synthesis. ModelSim also helps verify correctness before hardware testing, reducing debugging time.

Hardware peripherals demonstrate real-world use of protocols, aligning with stakeholder needs (students, educators, developers).

Scalability Plan:

- We can scale this project to design by understanding its fundamentals and in future we can implement this to create more complex design.
- This project could be extendable to implement more other advanced communication protocols.
- It could be documented for educational purpose or can be added to blog to provide information for this implementation to others.
- We can include some error detection or correction algorithms to it to make it more accurate for other application where this is useful.
- We can also convert one protocol data to another protocol data and develop cross communication protocols.

Conclusion:

This project will demonstrate the implementation of fundamental communication protocols such as UART, SPI, and I2C on FPGA platforms Xilinx Nexys 4 DDR. By using the modular design approach, we develop protocols in Verilog, Also we used built-in sensors on the FPGA board, such as the ADXL362 accelerometer and ADT7420 temperature sensor, highlights practical real-world interfacing which strengthens the educational value of the project.

Overall, the project will bridge the gap between theory and practice in digital system design, providing a hands-on learning platform for students, educators, and developers. It not only builds FPGA proficiency but also prepares learners for industry-level challenges in VLSI, embedded systems, and communication protocol design.

References:

https://si.farnell.com/communication-network-protocols-trc-ar
https://www.semanticscholar.org/paper/A-Review-on-Implementation-of-UART-using-Different-Dhanadravye-Thorat/a7c7761f88aa0854d9af7cec14608e885dab43c6
https://learn.sparkfun.com/tutorials/i2c/all
https://www.nxp.com/docs/en/application-note/AN10386.pdf
https://digilent.com/reference/programmable-logic/nexys-4-ddr/reference-manual
https://www.analog.com/media/en/technical-documentation/data-sheets/adxl362.pdf
https://www.analog.com/media/en/technical-documentation/data-sheets/adt7420.pdf