

Final Project Report

*High-Level Synthesis (HLS) Implementation
on Zynq Hardware*

Student:

Ali Akbar MohsenNejad

Course:

System-Level Design Laboratory

Contents

1. Introduction	3
2. Experiment 1: XOR Gate.....	3
3. Experiment 2: Arithmetic Logic Unit (ALU).....	4
4. Experiment 3: Register Bank	6
5. Experiment 4: Finite State Machine (FSM).....	6
6. Experiment 5: Single Interrupt	7
7. Experiment 6: Multi Interrupt.....	8
8. Conclusion.....	9

1. Introduction

This report presents the re-implementation and evaluation of six digital design experiments that were originally developed in SystemC. The designs are now migrated to the hardware design flow using Vivado HLS and deployed on the Xilinx Zynq-7000 platform.

For each experiment:

- The problem statement and objectives are introduced.
- The C++/HLS implementation and applied pragmas are explained.
- A testbench validates the design functionality.
- Finally, synthesis reports are analyzed, including resource usage (LUT, FF, BRAM), latency, and timing results.

Target Device: xc7z020clg400-1 (Zynq-7000)

2. Experiment 1: XOR Gate

Objective: Implement and simulate a simple logical XOR gate in C++ using HLS.

- Files: xor_gate.h, xor_gate.cpp, xor_gate_tb.cpp
- Testbench validates all four input combinations.
- Synthesis Results: Only 2 LUTs used, negligible delay, minimal hardware footprint.

```
3
4 == Vivado HLS Report for 'xor_gate'
5
6 * Date: Tue Aug 5 18:38:30 2025
7
8 * Version: 2019.1 (Build 2552052 on Fri May 24 15:28:33 MDT 2019)
9 * Project: xor_hls_project
10 * Solution: xor_solution
11 * Product family: zynq
12 * Target device: xc7z020-clg400-1
13
14
15
16 == Performance Estimates
17
18 + Timing (ns):
19   * Summary:
20   +-----+-----+-----+-----+
21   | Clock | Target| Estimated| Uncertainty|
22   +-----+-----+-----+-----+
23   |ap_clk | 10.00| 0.978| 1.25|
24   +-----+-----+-----+-----+
25
26 + Latency (clock cycles):
27   * Summary:
28   +-----+-----+-----+-----+
29   | Latency | Interval | Pipeline|
30   | min | max | min | max | Type |
31   +-----+-----+-----+-----+
32   | 0| 0| 0| 0| none |
33   +-----+-----+-----+-----+
34
35
36 == Utilization Estimates
37
38 * Summary:
39   +-----+-----+-----+-----+-----+
40   | Name | BRAM_18K| DSP48E| FF | LUT | URAM|
41   +-----+-----+-----+-----+-----+
42   | DSP | -| -| -| -| -|
43   | Expression | -| -| 0| 2| -|
44   | FIFO | -| -| -| -| -|
45   | Instance | -| -| -| -| -|
46   | Memory | -| -| -| -| -|
47   | Multiplexer | -| -| -| -| -|
48   | Register | -| -| -| -| -|
49   +-----+-----+-----+-----+-----+
50   | Total | 0| 0| 0| 2| 0|
51   +-----+-----+-----+-----+-----+
52   | Available | 280| 220| 106400| 53200| 0|
53   +-----+-----+-----+-----+-----+
54   | Utilization (%) | 0| 0| 0| ~0 | 0|
55
```

3. Experiment 2: Arithmetic Logic Unit (ALU)

Objective: Implement an 8-bit ALU supporting arithmetic and logical operations with condition flags.

- Inputs/Outputs: Two 8-bit inputs (A, B), 3-bit opcode, Result R (8-bit), Flags (C, Z, N, V)
- Supported Operations: ADD, SUB, AND, OR, XOR, NOT, INC
- Testbench: Validates overflow and borrow conditions.
- Synthesis Results: Single-cycle latency, clock ≈ 3.98 ns, very low resource usage.

```
Console Tasks Problems Executables Debugger Console
<terminated> (exit value: 0) ALU.Debug [C/C++ Application] csim.exe
A=255 (11111111) B=128 (10000000) op=0 R=127 (01111111) C=1 Z=0 N=0 V=1
A=255 (11111111) B=128 (10000000) op=1 R=127 (01111111) C=1 Z=0 N=0 V=0
A=255 (11111111) B=128 (10000000) op=2 R=128 (10000000) C=0 Z=0 N=1 V=0
A=255 (11111111) B=128 (10000000) op=3 R=255 (11111111) C=0 Z=0 N=1 V=0
A=255 (11111111) B=128 (10000000) op=4 R=127 (01111111) C=0 Z=0 N=0 V=0
A=255 (11111111) B=128 (10000000) op=5 R=0 (00000000) C=0 Z=1 N=0 V=0
A=255 (11111111) B=128 (10000000) op=6 R=0 (00000000) C=1 Z=1 N=0 V=0
A=10 (00001010) B=2 (00000010) op=0 R=12 (00001100) C=0 Z=0 N=0 V=0
A=10 (00001010) B=2 (00000010) op=1 R=8 (00001000) C=1 Z=0 N=0 V=0
A=10 (00001010) B=2 (00000010) op=2 R=2 (00000010) C=0 Z=0 N=0 V=0
A=10 (00001010) B=2 (00000010) op=3 R=10 (00001010) C=0 Z=0 N=0 V=0
A=10 (00001010) B=2 (00000010) op=4 R=8 (00001000) C=0 Z=0 N=0 V=0
A=10 (00001010) B=2 (00000010) op=5 R=245 (11110101) C=0 Z=0 N=1 V=0
A=10 (00001010) B=2 (00000010) op=6 R=11 (00001011) C=0 Z=0 N=0 V=0
A=150 (10010110) B=173 (10101101) op=0 R=67 (01000011) C=1 Z=0 N=0 V=1
A=150 (10010110) B=173 (10101101) op=1 R=233 (11101001) C=0 Z=0 N=1 V=0
A=150 (10010110) B=173 (10101101) op=2 R=132 (10000100) C=0 Z=0 N=1 V=0
A=150 (10010110) B=173 (10101101) op=3 R=191 (10111111) C=0 Z=0 N=1 V=0
A=150 (10010110) B=173 (10101101) op=4 R=59 (00111011) C=0 Z=0 N=0 V=0
A=150 (10010110) B=173 (10101101) op=5 R=105 (01101001) C=0 Z=0 N=0 V=0
A=150 (10010110) B=173 (10101101) op=6 R=151 (10010111) C=0 Z=0 N=1 V=0
A=2 (00000010) B=10 (00001010) op=0 R=12 (00001100) C=0 Z=0 N=0 V=0
A=2 (00000010) B=10 (00001010) op=1 R=248 (11111000) C=0 Z=0 N=1 V=0
A=2 (00000010) B=10 (00001010) op=2 R=2 (00000010) C=0 Z=0 N=0 V=0
A=2 (00000010) B=10 (00001010) op=3 R=10 (00001010) C=0 Z=0 N=0 V=0
A=2 (00000010) B=10 (00001010) op=4 R=8 (00001000) C=0 Z=0 N=0 V=0
A=2 (00000010) B=10 (00001010) op=5 R=253 (11111011) C=0 Z=0 N=1 V=0
A=2 (00000010) B=10 (00001010) op=6 R=3 (00000011) C=0 Z=0 N=0 V=0
```

```

=====
== Vivado HLS Report for 'alu_hls_ex'
=====

```

```

* Date:          Sun Aug 10 14:59:01 2025

* Version:       2019.1 (Build 2552052 on Fri May 24 15:28:33 MDT)
* Project:       ALU
* Solution:      ALU_Solution
* Product family: virtex7
* Target device: xc7vx485t-ffgl1157-1

```

```

=====
== Performance Estimates
=====

```

```

+ Timing (ns):

```

```

  * Summary:

```

	Clock	Target	Estimated	Uncertainty
ap_clk	10.00		3.981	1.25

```

+ Latency (clock cycles):

```

```

  * Summary:

```

Latency		Interval		Pipeline
min	max	min	max	Type
1	1	1	1	none

```

  + Detail:

```

```

    * Instance:
      N/A

```

```

    * Loop:
      N/A

```

```

=====
== Utilization Estimates
=====

```

```

* Summary:

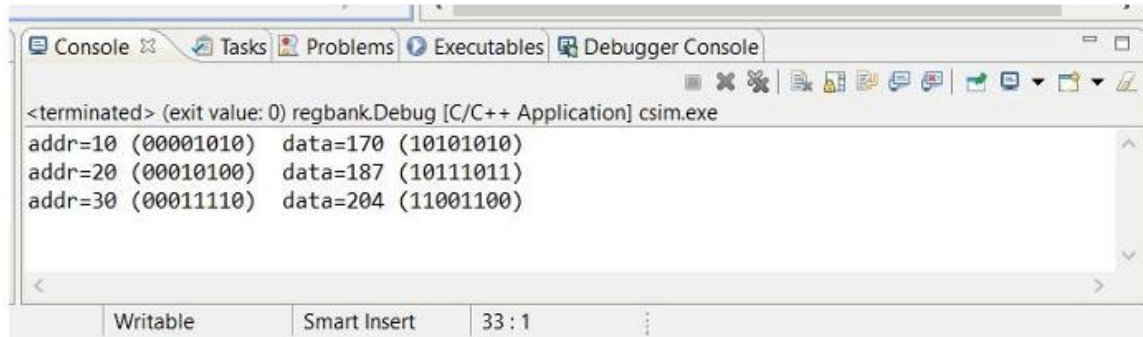
```

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	153	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	157	-
Register	-	-	2	-	-
Total	0	0	2	310	0
Available	2060	2800	607200	303600	0
Utilization (%)	0	0	~0	~0	0

4. Experiment 3: Register Bank

Objective: Implement a 256x8-bit memory block with read, write, and enable signals.

- Priority: Write takes precedence over read when enabled.
- Testbench: Writes values to addresses, then reads back correctly.
- Synthesis Results: Clock ≈ 2.27 ns, latency 1–2 cycles, minimal resources.

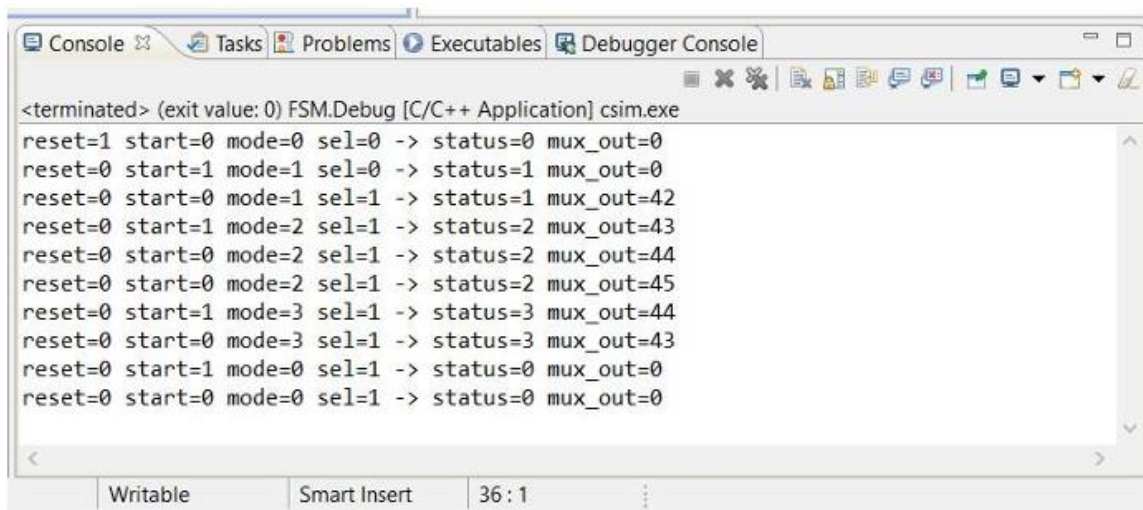


```
<terminated> (exit value: 0) regbank.Debug [C/C++ Application] csim.exe
addr=10 (00001010)  data=170 (10101010)
addr=20 (00010100)  data=187 (10111011)
addr=30 (00011110)  data=204 (11001100)
```

5. Experiment 4: Finite State Machine (FSM)

Objective: Design an FSM with states: IDLE, LOAD, COUNT_UP, COUNT_DOWN.

- Includes a 2:1 multiplexer for output selection.
- Testbench: Demonstrates transitions across states.
- Synthesis Results: Clock ≈ 5.12 ns, latency = 1 cycle, minimal resources.

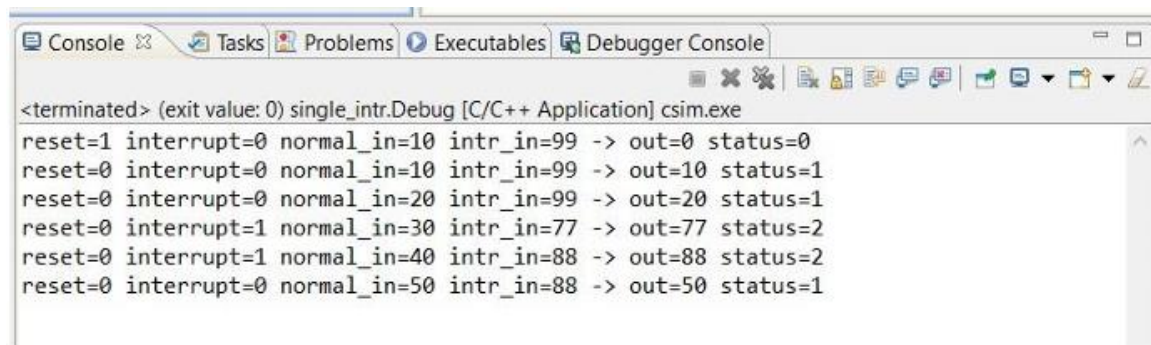


```
<terminated> (exit value: 0) FSM.Debug [C/C++ Application] csim.exe
reset=1 start=0 mode=0 sel=0 -> status=0 mux_out=0
reset=0 start=1 mode=1 sel=0 -> status=1 mux_out=0
reset=0 start=0 mode=1 sel=1 -> status=1 mux_out=42
reset=0 start=1 mode=2 sel=1 -> status=2 mux_out=43
reset=0 start=0 mode=2 sel=1 -> status=2 mux_out=44
reset=0 start=0 mode=2 sel=1 -> status=2 mux_out=45
reset=0 start=1 mode=3 sel=1 -> status=3 mux_out=44
reset=0 start=0 mode=3 sel=1 -> status=3 mux_out=43
reset=0 start=1 mode=0 sel=1 -> status=0 mux_out=0
reset=0 start=0 mode=0 sel=1 -> status=0 mux_out=0
```

6. Experiment 5: Single Interrupt

Objective: Implement a module reacting to a single interrupt signal.

- Modes: Normal \rightarrow output=in_normal, Interrupt \rightarrow output=in_intr.
- States: 0=IDLE, 1=Normal, 2=ISR
- Testbench: Reset \rightarrow Normal \rightarrow Interrupt \rightarrow Back to Normal.
- Synthesis Results: Clock \approx 1.24 ns, latency=0 cycles, only 23 LUTs.



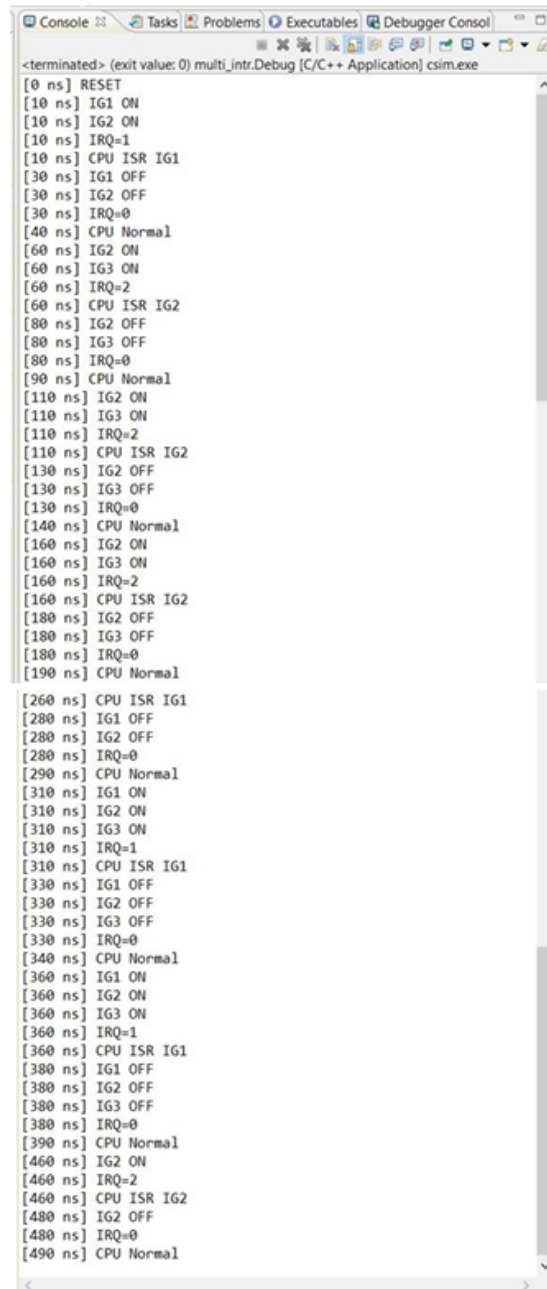
The screenshot shows a debugger console window with the following tabs: Console, Tasks, Problems, Executables, and Debugger Console. The console output is as follows:

```
<terminated> (exit value: 0) single_intr.Debug [C/C++ Application] csim.exe
reset=1 interrupt=0 normal_in=10 intr_in=99 -> out=0 status=0
reset=0 interrupt=0 normal_in=10 intr_in=99 -> out=10 status=1
reset=0 interrupt=0 normal_in=20 intr_in=99 -> out=20 status=1
reset=0 interrupt=1 normal_in=30 intr_in=77 -> out=77 status=2
reset=0 interrupt=1 normal_in=40 intr_in=88 -> out=88 status=2
reset=0 interrupt=0 normal_in=50 intr_in=88 -> out=50 status=1
```


7. Experiment 6: Multi Interrupt

Objective: Implement a multi-interrupt controller with 3 sources (IG1, IG2, IG3) and fixed priority (IG3>IG2>IG1).

- Behavior: Each interrupt ON for 20ns, OFF for 30ns.
- CPU services ISR for 30ns, then returns to Normal.
 - Outputs: irq, cpu_state, ig1, ig2, ig3
- Testbench: 500 ns simulation confirms priority and ISR timing.
- Synthesis Results: Clock \approx 4.39 ns, latency=1–2 cycles, \sim 400 LUTs, 70 FFs.



```
<terminated> (exit value: 0) multi_intr.Debug [C/C++ Application] csim.exe
[0 ns] RESET
[10 ns] IG1 ON
[10 ns] IG2 ON
[10 ns] IRQ=1
[10 ns] CPU ISR IG1
[30 ns] IG1 OFF
[30 ns] IG2 OFF
[30 ns] IRQ=0
[40 ns] CPU Normal
[60 ns] IG2 ON
[60 ns] IG3 ON
[60 ns] IRQ=2
[60 ns] CPU ISR IG2
[80 ns] IG2 OFF
[80 ns] IG3 OFF
[80 ns] IRQ=0
[90 ns] CPU Normal
[110 ns] IG2 ON
[110 ns] IG3 ON
[110 ns] IRQ=2
[110 ns] CPU ISR IG2
[130 ns] IG2 OFF
[130 ns] IG3 OFF
[130 ns] IRQ=0
[140 ns] CPU Normal
[160 ns] IG2 ON
[160 ns] IG3 ON
[160 ns] IRQ=2
[160 ns] CPU ISR IG2
[180 ns] IG2 OFF
[180 ns] IG3 OFF
[180 ns] IRQ=0
[190 ns] CPU Normal
[260 ns] CPU ISR IG1
[280 ns] IG1 OFF
[280 ns] IG2 OFF
[280 ns] IRQ=0
[290 ns] CPU Normal
[310 ns] IG1 ON
[310 ns] IG2 ON
[310 ns] IG3 ON
[310 ns] IRQ=1
[310 ns] CPU ISR IG1
[330 ns] IG1 OFF
[330 ns] IG2 OFF
[330 ns] IG3 OFF
[330 ns] IRQ=0
[340 ns] CPU Normal
[360 ns] IG1 ON
[360 ns] IG2 ON
[360 ns] IG3 ON
[360 ns] IRQ=1
[360 ns] CPU ISR IG1
[380 ns] IG1 OFF
[380 ns] IG2 OFF
[380 ns] IG3 OFF
[380 ns] IRQ=0
[390 ns] CPU Normal
[460 ns] IG2 ON
[460 ns] IRQ=2
[460 ns] CPU ISR IG2
[480 ns] IG2 OFF
[480 ns] IRQ=0
[490 ns] CPU Normal
```


8. Conclusion

All six experiments were successfully migrated from SystemC (software simulation) to Vivado HLS (hardware design). The designs synthesized on Zynq-7000 with minimal resource utilization and very low latencies. This demonstrates effective use of C++/HLS for hardware-software co-design.