Lab 7: High Level Synthesis (Xilinx VIVADO HLS)

ENGG3050: Reconfigurable Computing Systems

Instructor:
Dr. Shawki Areibi

Group 42: Tuesday-3:30 Section

Bilal Ayyache: 0988616
Anthony Granic: 0994824

Lab Start Date: November 7, 2019
Lab End Date: November 28, 2019

Contents

List of Figures

1 Tutorial 1, 2 and 3 Questions

Tutorial 1:
Question 1:
- Estimated clock period: 6.38 ns
- Worst case latency: 106 clock cycles
- Number of DSP48E used: 1
- Number of FFs used: 61
- Number of LUTs used: 52

Tutorial 2:
Question 1:
- Estimated clock period: 11.19ns
- Worst case latency: 56536325
- Number of DSP48E used: 12
- Number of BRAMs used: 12288
- Number of FFs used: 219
- Number of LUTs used: 760

Question 2:
- Estimated clock period: 8.34ns
- Worst case latency: 19664641
- Number of DSP48E used: 5
- Number of FFs used: 198
- Number of LUTs used: 253

Question 3:
- Estimated clock period: 11.19ns
- Worst case latency: 19664641
- Number of DSP48E used: 4
- Number of FFs used: 203
- Number of LUTs used: 238

1.1 Directives and Speedup table

Directives	CLK CYC	DSPs	FFs	LUTs	Timing	Speedup
Baseline	106	1	61	52	6.38	100%
A(UNROLL)	9	27	538	92	9.4	-87.4904
B(PIPELINE)	9	27	663	66	8.77	-88.3288
C(DATAFLOW)	106	1	62	52	6.38	0
D(INLINE)	106	1	61	52	6.38	0
E(PARTITION)	106	1	62	112	8.77	37.46082
AE	5	27	494	91	9.4	-93.0502
BE	5	27	597	92	9.4	-93.0502
CE	106	1	63	112	8.77	37.46082
DE	106	1	62	112	8.77	37.46082
ABE	5	27	597	92	9.4	-93.0502
ACE	5	27	495	91	9.4	-93.0502
ADE	5	27	494	91	9.4	-93.0502
BCE	106	1	63	112	8.77	37.46082
BDE	5	27	597	92	9.4	-93.0502
CDE	106	1	63	112	8.77	37.46082
ABCE	5	27	495	91	9.4	-93.0502
ABDE	5	27	597	92	9.4	-93.0502
ACDE	5	27	495	91	9.4	-93.0502
BCDE	106	1	63	112	8.77	37.46082
ABCDE	5	27	495	91	9.4	-93.0502

1.2 Conclusion

Conclusion In this table 5 types of directives were applied to the code in different combinations. These directives were loop unrolling, pipelining, dataflow, inline, and partitioning. Applying these in different combinations made the system either speedup or slow down. The highest speedup was a gain of 37.5%, achieved by multiple combinations of directives. The biggest loss in speed was 93.1%, also achieved by multiple combinations of directives. The usages for each implementation were relatively similar for DSPs and LUTS. The amount of flip flops varied greatly between different implementations. In conclusion it is very useful to explore multiple designs before deciding on the best one in terms of speedup and board usage.

2 Design 3

2.1 Implementation

Design 3 introduces various techniques and directives which can be used in Vivado HLS to improve design performance as well as area and resource utilization. The design under consideration performs discrete cosine transformation (DCT) on an 8x8 block of data.

2.2 Directives used for final solution

Pipeline, Dataflow, Inline, and reshape directives were used to decrease latency. PIPELINE directive when applied to outer loop causes the inner loop to unroll. When a loop is unrolled, resources utilization increases as operations are done concurrently. Partitioning memory improves performance but in return increases BRAM utilization. When INLINE directive is applied to a function, the lower level hierarchy is automatically dissolved. When DATAFLOW directive is applied, the default memory buffers are automatically inserted between the top-level functions and loops. The RESHAPE directive will allow multiple accesses to BRAM, however, care should be taken if a single element requires modification as it will result in read-modify-write operation for the entire word.

2.3 C Program Used

```
#include "dct.h"

void dct_1d(dct_data_t src[DCT_SIZE], dct_data_t dst[DCT_SIZE])
{
    unsigned int k, n;
    int tmp;
    const dct_data_t dct_coeff_table[DCT_SIZE][DCT_SIZE] = {
#include "dct_coeff_table.txt"
};
DCT_Outer_Loop:
    for (k = 0; k < DCT_SIZE; k++) {
DCT_Inner_Loop:
        for(n = 0, tmp = 0; n < DCT_SIZE; n++) {
            int coeff = (int)dct_coeff_table[k][n];
            tmp += src[n] * coeff;
        }
        dst[k] = DESCALE(tmp, CONST_BITS);
    }
}
```

```
void dct_2d(dct_data_t in_block[DCT_SIZE][DCT_SIZE],
      dct_data_t out_block[DCT_SIZE][DCT_SIZE])
{

   dct_data_t row_outbuf[DCT_SIZE][DCT_SIZE];
   dct_data_t col_outbuf[DCT_SIZE][DCT_SIZE], col_inbuf[DCT_SIZE][DCT_SIZE];
   unsigned i, j;

   // DCT rows
Row_DCT_Loop:
   for(i = 0; i < DCT_SIZE; i++) {
      dct_1d(in_block[i], row_outbuf[i]);
   }
   // Transpose data in order to re-use 1D DCT code
Xpose_Row_Outer_Loop:
   for (j = 0; j < DCT_SIZE; j++)
Xpose_Row_Inner_Loop:
      for(i = 0; i < DCT_SIZE; i++)
         col_inbuf[j][i] = row_outbuf[i][j];
   // DCT columns
Col_DCT_Loop:
   for (i = 0; i < DCT_SIZE; i++) {
      dct_1d(col_inbuf[i], col_outbuf[i]);
   }
   // Transpose data back into natural order
Xpose_Col_Outer_Loop:
   for (j = 0; j < DCT_SIZE; j++)
Xpose_Col_Inner_Loop:
      for(i = 0; i < DCT_SIZE; i++)
         out_block[j][i] = col_outbuf[i][j];
}

void read_data(short input[N], short buf[DCT_SIZE][DCT_SIZE])
{
   int r, c;

RD_Loop_Row:
   for (r = 0; r < DCT_SIZE; r++) {
RD_Loop_Col:
      for (c = 0; c < DCT_SIZE; c++)
```

```
            buf[r][c] = input[r * DCT_SIZE + c];
    }
}

void write_data(short buf[DCT_SIZE][DCT_SIZE], short output[N])
{
    int r, c;

WR_Loop_Row:
    for (r = 0; r < DCT_SIZE; r++) {
WR_Loop_Col:
        for (c = 0; c < DCT_SIZE; c++)
            output[r * DCT_SIZE + c] = buf[r][c];
    }
}

void dct(short input[N], short output[N])
{

    short buf_2d_in[DCT_SIZE][DCT_SIZE];
    short buf_2d_out[DCT_SIZE][DCT_SIZE];

    // Read input data. Fill the internal buffer.
    read_data(input, buf_2d_in);

    dct_2d(buf_2d_in, buf_2d_out);

    // Write out the results.
    write_data(buf_2d_out, output);
}
```

2.4 Results Analysis

2.4.1 Program performance before improvements:

- Estimated clock period: 6.508
- Worst case latency: 3959
- Number of DSP48E used: 1
- Number of BRAMs used: 5
- Number of FFs used: 278
- Number of LUTs used: 982

Performance Estimates

- **Timing (ns)**
  - **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	6.508	1.25

- **Latency (clock cycles)**
  - **Summary**

Latency		Interval		Type
min	max	min	max	
3959	3959	3959	3959	none

  - **Detail**
    - **Instance**

Instance	Module	Latency		Interval		Type
		min	max	min	max	
grp_dct_2d_fu_152	dct_2d	3668	3668	3668	3668	none

    - **Loop**

Loop Name	Latency		Iteration Latency	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
- RD_Loop_Row	144	144	18	-	-	8	no
+ RD_Loop_Col	16	16	2	-	-	8	no
- WR_Loop_Row	144	144	18	-	-	8	no
+ WR_Loop_Col	16	16	2	-	-	8	no

Figure 1: Performance Before Improvements

Utilization Estimates

- **Summary**

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	156	-
FIFO	-	-	-	-	-
Instance	3	1	209	677	0
Memory	2	-	0	0	0
Multiplexer	-	-	-	149	-
Register	-	-	69	-	-
Total	5	1	278	982	0
Available	280	220	106400	53200	0
Utilization (%)	1	~0	~0	1	0

Figure 2: Utilization Before Improvements

2.4.2 Program performance after improvements:

- Estimated clock period: 9.400
- Worst case latency: 277
- Number of DSP48E used: 96
- Number of BRAMs used: 10
- Number of FFs used: 3884
- Number of LUTs used: 3729

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	9.400	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
648	648	298	298	dataflow

Figure 3: Performance After Improvements

Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	2	-
FIFO	-	-	-	-	-
Instance	0	16	1639	3267	-
Memory	8	-	0	0	0
Multiplexer	-	-	-	-	-
Register	-	-	-	-	-
Total	8	16	1639	3269	0
Available	280	220	106400	53200	0
Utilization (%)	2	7	1	6	0

Figure 4: Utilization After Improvements

2.5 Lab Feedback

Lab implementation was successful. Lab requirements in the lab description document was not clear. The lab mentions to go through tutorial 3 but does not exactly state was needs to be reported from implementing tutorial. Pipelining analysis and loop performance analysis was not included due to unclear requirements. In previous labs, marks were deducted for reporting details not asked for, an example is lab 1 where a subtraction functionality was added. It was hard to understand what is required to report.

3 Design 4

3.1 Introduction

This lab introduces a design flow to generate a IP-XACT adapter from a FIR design (provided by Dr.Shawki) using Vivado HLS and using the generated IP-XACT adapter in a processor system using IP Integrator in Vivado.

3.2 System Overview

A peripheral core of the designed filter was developed. The core was instantiated in the processor system of the ZED-Board. The processor system acquired a stereo music stream using an on-board CODEC chip and I2C controller, and processed through the designed filter (bandstop filter). The filtered soundtrack was then outputted to a headset connected to the board.

when creating the project, a C simulation was first processed to analyze latency and usage. The design was later synthesized to ensure functionality. An RTL/C co-simulation was executed and results were analyzed. An IP-XACT adapter was generated and used to facilitate the 2 FIR filters required to filter the disrupted soundtrack.

The design was exported to SDK to create an application and test the implementation. The final soundtrack was filtered successfully.

3.3 Hardware Implementation

3.3.1 Block Design

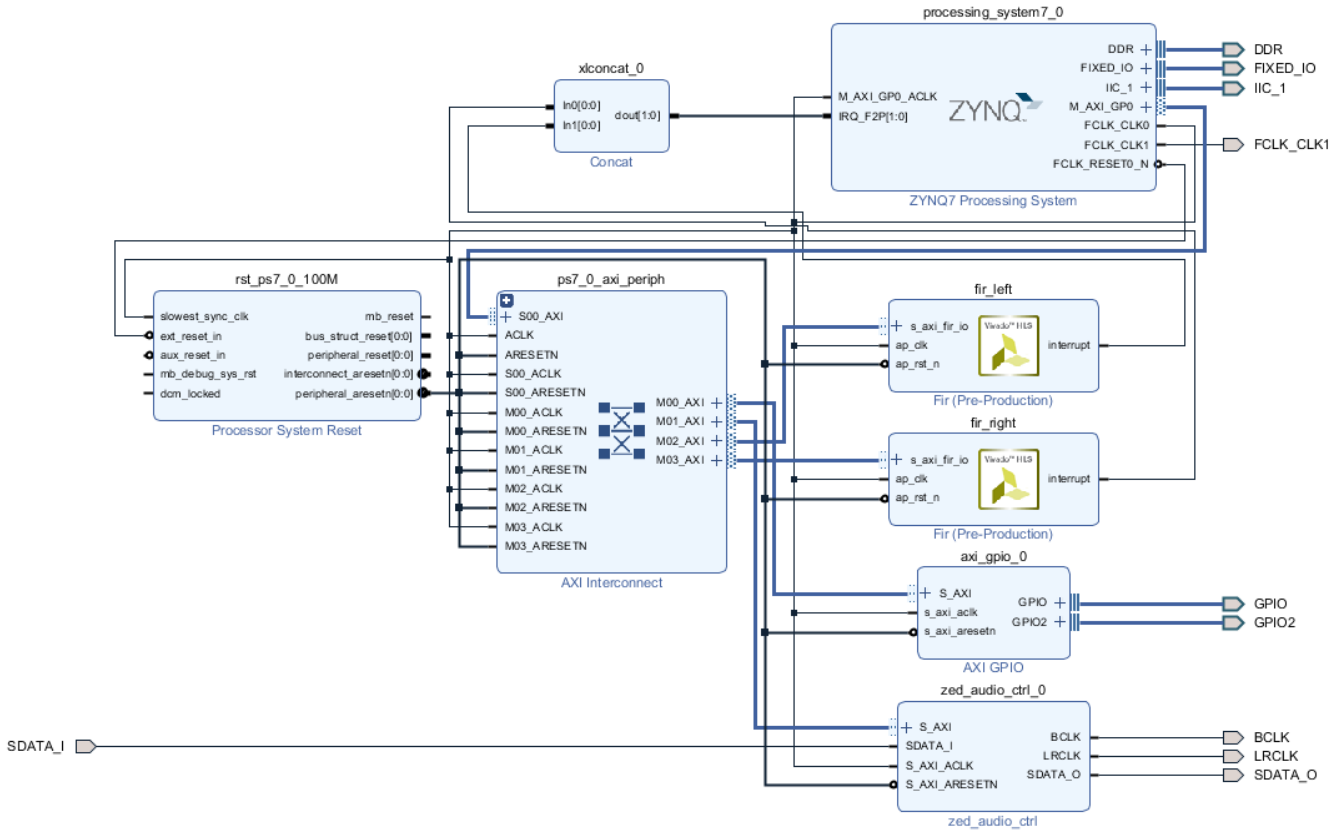

Figure 5: System's Block Design

3.3.2 Usage Report

Name	Slice LUTs (53200)	Slice Registers (106400)	Slice (13300)	LUT as Logic (53200)	LUT as Memory (17400)	Block RAM Tile (140)
system_wrapper	1073	1270	420	963	110	1270
system_i (system)	1073	1270	420	963	110	0

Figure 6: Resources Used

3.3.3 Timing Analysis

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	3.793 ns	Worst Hold Slack (WHS):	0.039 ns	Worst Pulse Width Slack (WPWS):	3.750 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	3639	Total Number of Endpoints:	3639	Total Number of Endpoints:	1392

Figure 7: Timing Analysis Summary

3.3.4 Implemented Design

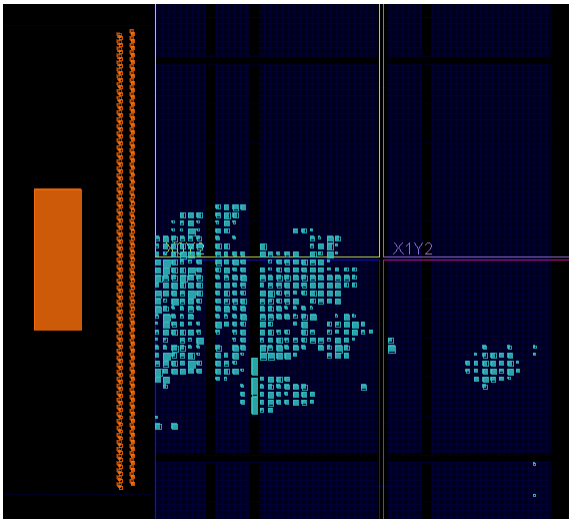

Figure 8: Hardware Utilization

3.4 Software Implementation

3.4.1 Implementation C-Program

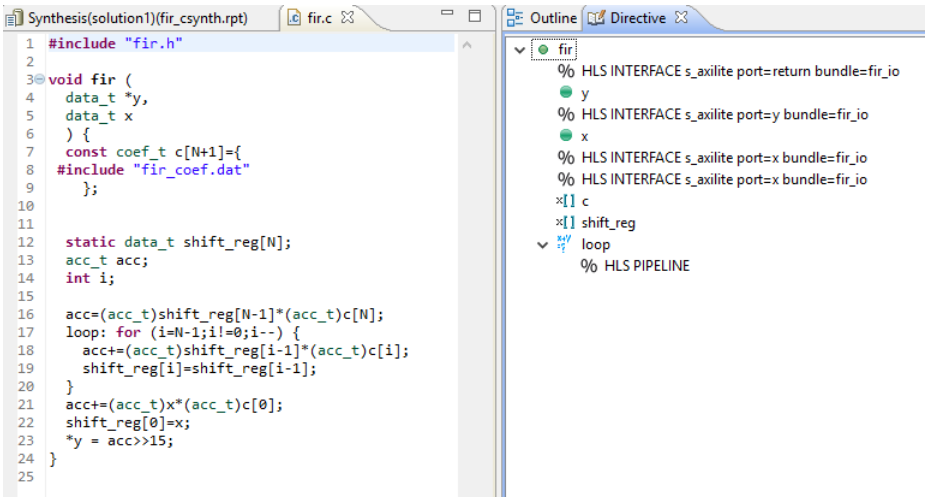

Figure 9: C-Program Used For Implementation

3.4.2 Software Performance

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.702	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
62	62	62	62	none

Figure 10: Performance Estimates Of C-Program

3.4.3 Software Utilization

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	3	-	-	-
Expression	-	-	0	30	-
FIFO	-	-	-	-	-
Instance	0	-	80	104	-
Memory	0	-	48	30	0
Multiplexer	-	-	-	123	-
Register	-	-	139	-	-
Total	0	3	267	287	0
Available	280	220	106400	53200	0
Utilization (%)	0	1	~0	~0	0

Figure 11: Utilization Summary for C-Program

3.4.4 Pipelining Directory

Properties:	Before Pipelining	After Pipelining
Estimated clock period:	8.702	8.702
Worst case latency:	174	62
Number of DSP48E used:	3	3
Number of BRAMs used:	0	0
Number of FFs used:	167	267
Number of LUTs used:	154	287

Figure 12: Performance Before And After Pipelining

3.5 Lab Feedback

Lab implementation was successful. Soundtrack was filtered successfully. Some problems were faced during implementation, but were solved later on in the design process. The first problem was insuring that the design was placed in the same repository mentioned zed_audio_project_create.tcl . In this design line 19 was changed to set_property ip_repo_paths C:/Users/bilal/lab7Part2/ip_repo [current_project] . The second problem faced was with the zed_audio_constraints.xdc . The xdc file does not declare all

pins which restricted Vivado from creating a .bit file. To solve this problem insert set_property BITSTREAM.General.UnconstrainedPins Allow [current_design] at the top of the xdc file. After these changes the lab was successfully implemented