ELEC6234 Embedded Processor Synthesis

Coursework

“SystemVerilog Design of an Application Specific Embedded Processor ”

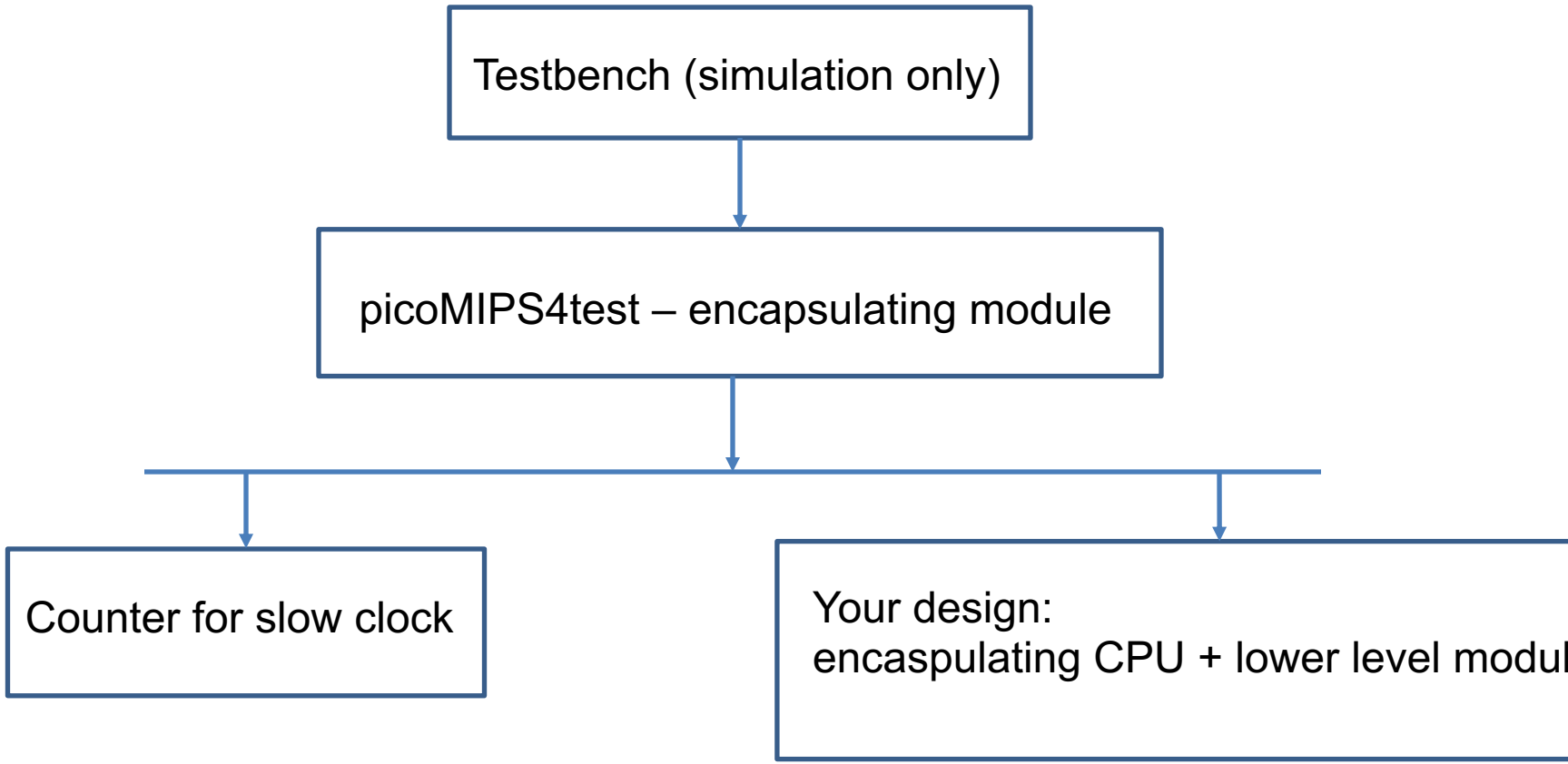
Introduction

- This exercise is done individually and the assessment is:
  - By formal report describing the final design, its development, implementation and testing.
  - By a laboratory demonstration of the final design on an Altera FPGA development system
- Assessment – 50% of the module

Design cost competition

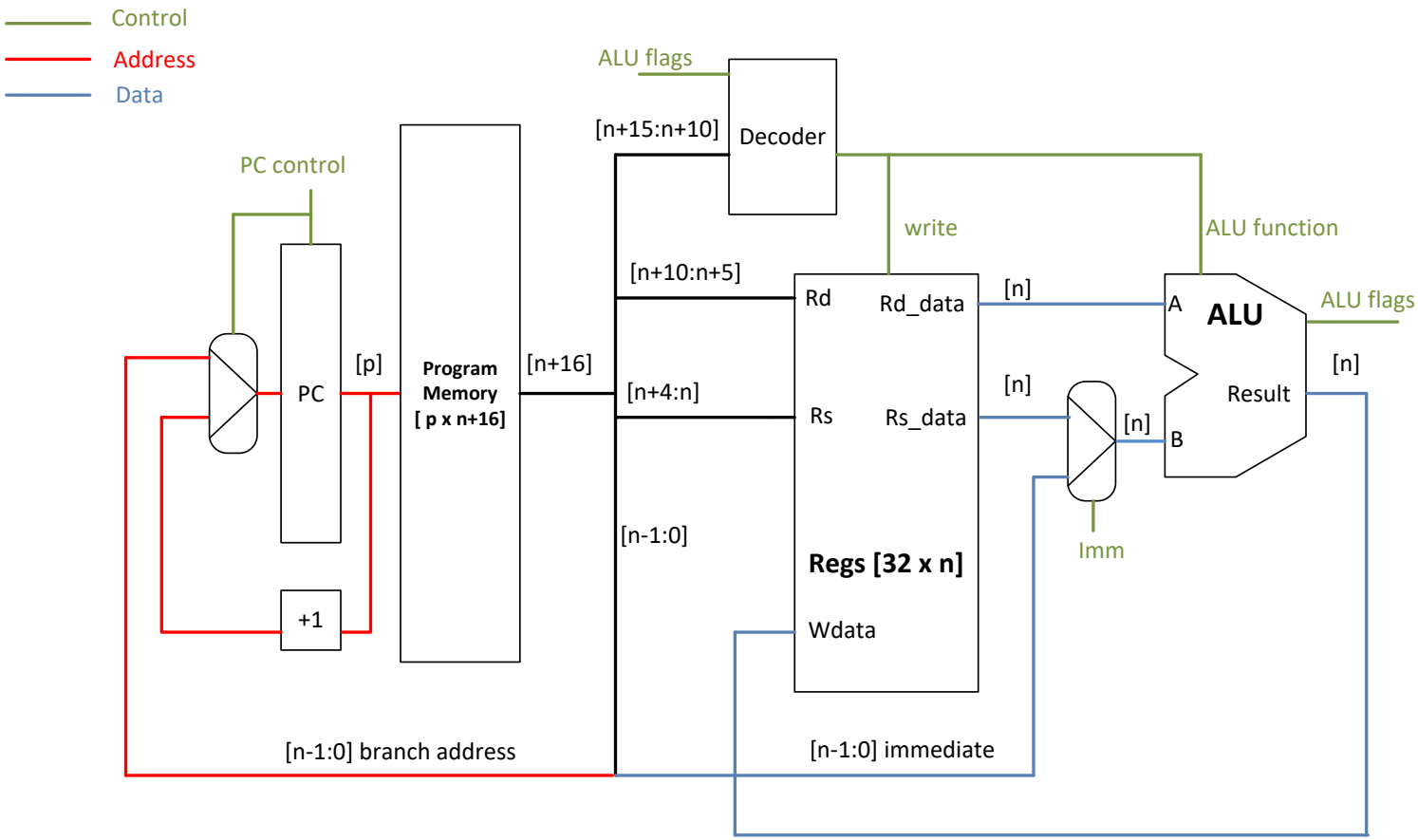
- The design should be as small as possible in terms of FPGA resources but sufficient to implement the specified affine transformation algorithm.
- *Cost = number of Adaptive Logic Modules (ALMs) used + max(0,number of embedded multipliers used – 2) +30 x Kbits of RAM used*
- Each ALM has 2 flip-flops hence flip-flops are included in the above cost figure.
- The cost figure should be calculated for an Altera Cyclone V SoC 5CSEMA5F31C6 device and should be as low as possible.
- Altera Cyclone V has 174 18x18 bit (or 348 9x9 bit) embedded hardware multipliers.
- To demonstrate the cost figure of your design show in your report Altera Quartus synthesis statistics for Cyclone V SoC 5CSEMA5F31C6 .

Design structure

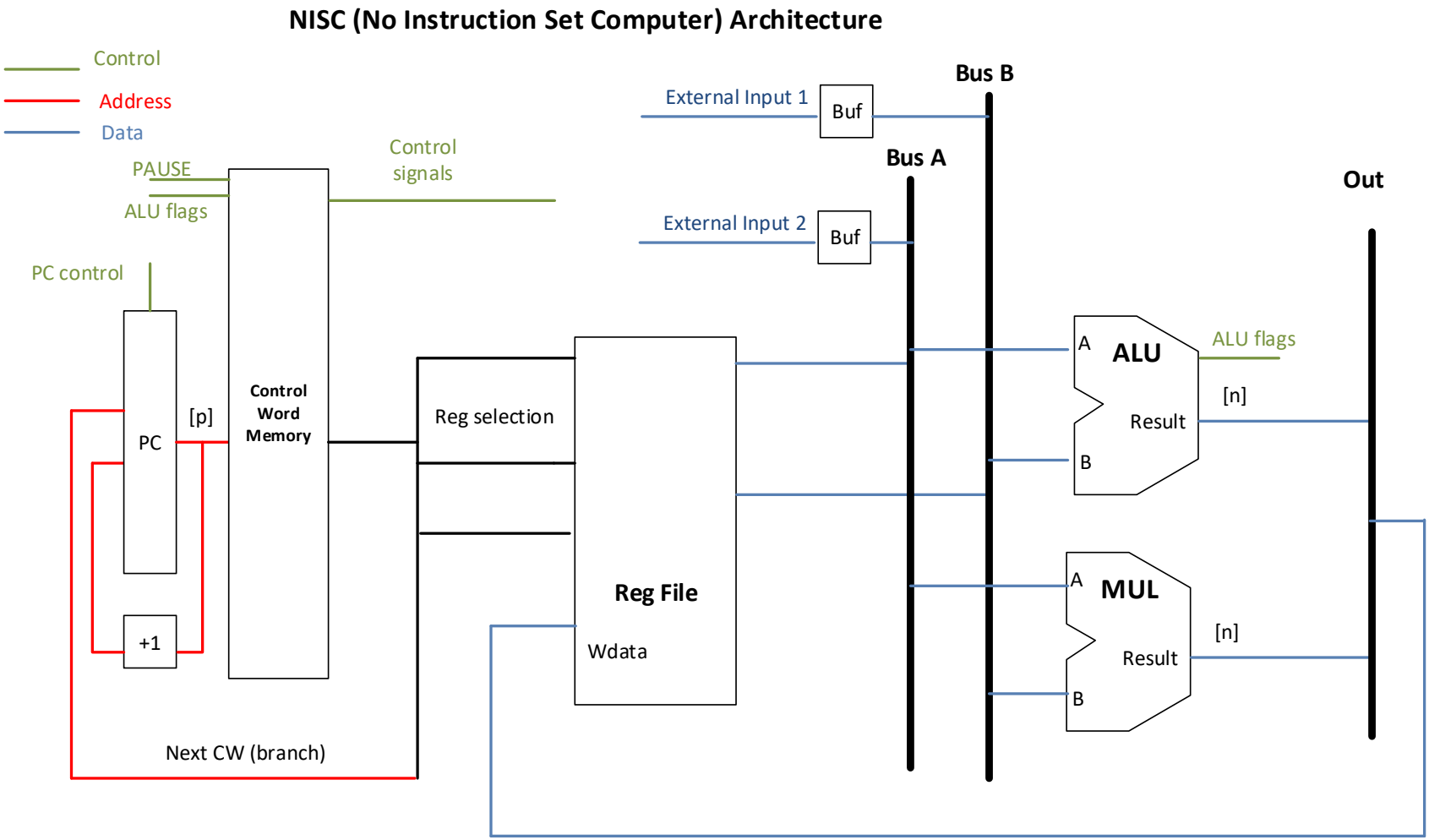


Use provided code for picoMIPS4test and Counter

picoMIPS


Control
Address
Data
ALU flags
[n+15:n+10]
Decoder
PC control
write
ALU function
[n+10:n+5]
Rd
Rd_data
[n]
A
ALU
ALU flags
[p]
Program
Memory
[p x n+16]
[n+16]
PC
[n+4:n]
Rs
Rs_data
[n]
[n]
Result
[n]
B
[n-1:0]
Imm
Regs [32 x n]
+1
Wdata
[n-1:0] branch address
[n-1:0] immediate

You are free to modify and adapt the picoMIPS architecture



Affine transformation applied to a single pixel

- An affine transformation is a geometrical transformation that preserves co-linearity, i.e. all points lying on a line will also lie on a line after the transformation and distance ratios are preserved.
- For two-dimensional images, the general affine transformation can be expressed as:

$$\begin{vmatrix} x_2 \\ y_2 \end{vmatrix} = A \times \begin{vmatrix} x_1 \\ y_1 \end{vmatrix} + B$$

- For example, a pure translation of pixels occurs if:

$$A = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}, B = \begin{vmatrix} b_1 \\ b_2 \end{vmatrix}$$

- The following coefficients implement pure scaling:

$$A = \begin{vmatrix} a_{11} & 0 \\ 0 & a_{22} \end{vmatrix}, B = \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$

Two sets of affine transformation coefficients to chose from

- In your implementation, choose one of the two following data sets

$$A = \begin{bmatrix} 0.75 & 0.5 \\ -0.5 & 0.75 \end{bmatrix} \qquad B = \begin{bmatrix} 20 \\ -20 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.5 & -0.875 \\ -0.875 & 0.75 \end{bmatrix} \qquad B = \begin{bmatrix} 5 \\ 12 \end{bmatrix}$$

- They both represent rotation, scaling and translation combined into a single affine transformation.

picoMIPS Implementation

- Pixel coordinates must be read from the switches SW0-SW7 on the FPGA development system and the resulting pixel coordinates after the transformation displayed on the LEDS LED0-LED7. Switch SW8 provides handshaking functionality as described in the pseudocode below. Switch SW9 should act as an active low reset.

- **Pseudocode:**
- Wait for coordinate *x1* by polling switch SW8. Wait while SW8=0. When SW8 becomes 1 (SW8=1) read the coordinate *x1* from SW0-SW7.
- Wait for switch SW8 to become 0
- Wait for coordinate *y1* as specified in step 1.
- Wait for SW8 to become 0
- Execute the affine transformation algorithm and display coordinate *x2* on LED0-LED7.
- Wait until SW8 becomes 1
- Display coordinate *y2* on LED0-LED7.
- Wait until SW8 becomes 0
- Go to step 1.

Input/Output

- The input/output functionality can be implemented in several different ways.
- For example, you can design your own IN and OUT instructions for reading/writing data using external ports.
- To use fewer hardware resources, you can consider connecting the ALU output, or a register output directly to the LEDs.
- You could consider dedicating a specific register number, e.g. register 1 to the input port.
  - In this way, an ADD instruction can be used to read data, e.g. ADD %5, %0, %1 would store the input data in register %5. Be creative and use your imagination!

Suggested data formats

- Coefficient of matrix A or input samples [x1,y1] and results [x2,y2] are 2's complement signed integers. Coefficients of A are fixed-point fractions in the range 0 .. +1 – 2^(-8), i.e. 2's complement numbers with the radix point positioned after the most significant bit.
- Therefore the weights of the individual bits are:

Bit position	Weight
7 (MSB)	- 2^0
6	2^(-1)
5	2^(-2)
4	2^(-3)
3	2^(-4)
2	2^(-5)
1	2^(-6)
0 (LSB)	2(-7)

- When fractional numbers above are multiplied by 2's complement integers, a double-length 16-bit product is obtained which is a 2's complement number with the radix point positioned after the 9-th bit. Note however that the output results must be 8-bit 2's complement whole numbers.

Binary multiplication examples

- Binary multiplication of 2's complement 8-bit numbers yields a 16-bit result. As one of the numbers is represented in the range -1..+1-2^{-7} and the other in the range -128..127, it is important to determine correctly which 8-bits of the 16-bit result represent the integer part which should be used for further calculations. The following examples illustrate which result bits represent the integer part.
- **Example 1. Multiply 0.75 x 6.**

weights:	-2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
0.75 =	0.	1	1	0	0	0	0	0

weights:	-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
6 =	0	0	0	0	0	1	1	0.

The 16-bit result is:

-2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
0	0	0	0	0	0	1	0	0.	1	0	0	0	0	0	0

- which represents the value of 4.5. The shaded area shows which bits need to be extracted when the representation is truncated to 8 bits.
- Note that the fraction part is discarded entirely, so the 8-bit result is now 4.
- Also note that when the leading bit is discarded, the weight of the new leading bit must now change from 2^7 to -2^7. Why?

Binary multiplication examples…

- The importance of the correct interpretation of the leading bit's weight is evident in the following example, where the result is negative.
- **Example 2. Multiply -0.25 x 20.**

weights:	-2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
-0.25 =	1.	1	1	0	0	0	0	0

weights:	-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
20 =	0	0	0	1	0	1	0	0.

The 16-bit result is:

-2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
1	1	1	1	1	1	0	1	1.	0	0	0	0	0	0	0

which is -5 in decimal representation.

- Again, the shaded area shows which 8 bits to extract when the result is truncated from 16 to 8 bits for further calculations.
- The truncated 8-bit result has the weight of -2^7 on the most significant bit so that it still correctly represents -5:

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	1	1	0	1	1.

Design strategy

- Develop SystemVerilog code and a separate testbench for each module in your design.
- Simulate each module in Modelsim.
- Synthesise each module in Altera Quartus and carefully analyse the synthesis warnings, statistics and RTL diagrams.
- When you are satisfied that all your modules are correct, write a testbench for the whole design and simulate.
- Synthesise the whole design and again, carefully analyse the warnings, statistics and RTL diagrams.
- You will be able to take an FPGA Development System on loan
  - Test your design either at home or in the laboratory.
- After the Easter Break you will be asked to demonstrate your design in the Electronics Laboratory.

Formal report

- Submit an electronic copy of your report through the electronic handin system, and a printed copy to the ECS front office by the deadline specified on the ELEC6234 notes website.
- The report should follow the report template provided
- It must contain a description of your design, including the final circuit diagrams, your instruction set and your program.
- Source files must be packaged in a zip file and submitted electronically as a separate file at the same time.
- 20% of the marks are allocated to the report, its style and organisation, 80% for the technical content.
- Bonus marks are awarded for implementation of novel concepts.