# COL216 Computer Architecture

## Lab Assignments 1 to 3

The objective of the lab assignments 1 to 3 is to design and implement a simple arithmetic expression evaluator in ARM assembly language. ARMSim# simulator is to be used for developing and testing the assembly programs.

| Week | Title and brief description |
|---|---|
| 1 | **Evaluation of expressions without parentheses**<br><br>Expression is represented as ASCII string<br><br>Operators add (+), subtract (-) and multiply (*): left to right evaluation, no operator precedence<br><br>Constant integer operands – ASCII to binary conversion required |
| 2 | **Evaluation of expressions with nested parentheses**<br><br>Use recursive functions to take care of nesting of parentheses |
| 3 | **Expression evaluation with input/output**<br><br>Use input/output facilities of Embest board plug-in of ARMSim# simulator |

# COL216 Computer Architecture

## Lab Assignments 4 to 14

The objective of the lab assignments 4 onwards is to design and implement a complete ARM CPU based computer, using Vivado tools and BASYS3 FPGA boards. These assignments are organized as follows.

| Week | Title and brief description |
|---|---|
| 4 | **CPU for a small ARM instructions subset**<br>Instructions for implementation = {add, sub, cmp, mov, ldr, str, beq, bne, b}<br>Single-cycle design style |
| 5 | **Simulate CPU design and prepare for synthesis**<br>Use ROM, RAM generator, ROM with pre-loaded test program<br>Design a test bench and do extensive simulation<br>Create interface for displaying address, instruction and result |
| 6 | **Design for synthesis and testing on the board**<br>Introduce FSM for single-step and continuous operation<br>Initialize ROM with multiple test programs, selectable at reset time<br>Study resource usage and performance |
| 7 | **CPU with multi-cycle design style**<br>Separate RF and ALU structurally<br>Extend FSM for -<br>multi-cycle execution<br>single-instruction operation, in addition to single-step and continuous<br>Study resource usage and performance |
| 8 | **CPU with all ARM DP instructions**<br>Extend the design to include all 16 DP instructions<br>Include all DP variants (flag setting, shift, rotate) |
| 9 | **CPU with all ARM DT instructions** |

| | |
|---|---|
| | Include all DT data types (byte, half word, signed byte/half word)<br>Include all DT variants (defined by PUW) |
| 10 | **CPU with all ARM multiply instructions**<br>Include all multiply instructions – short as well as long |
| 11 | **CPU with predication**<br>Test all conditions defined by the *condition* field<br>Apply predication to all instructions along with branch instruction |
| 12 | **CPU with interrupts**<br>Include SWI instruction and HW interrupts |
| 13 | **CPU with serial I/O**<br>Interface UART |
| 14 | **CPU with programming environment**<br>Use a simple monitor program to –<br>load and run user programs from terminal<br>provide simple I/O for user programs |