Design of a CORDIC unit architecture

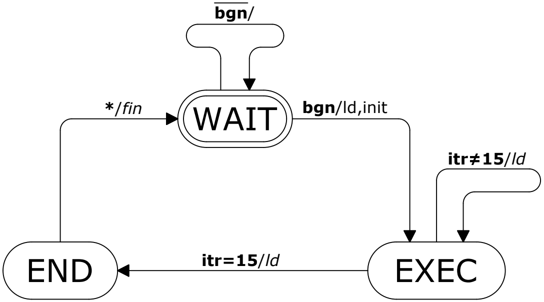
Proposed problems

Oprițoiu Flavius
flavius.opritoiu@cs.upt.ro

December 4, 2022

Problem 1

Design the *cordicctrl* FSM, given its state transition diagram:

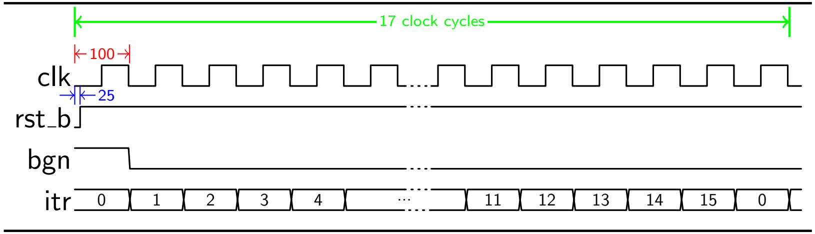


The interface of the module will be:

```
module cordicctrl(
  input clk,rst_b,bgn,
  input [3:0] itr,
  output ld,init,fin
);
```

Problem 1 (contd.)

Construct a testbench generating the inputs as in the timing diagram bellow:

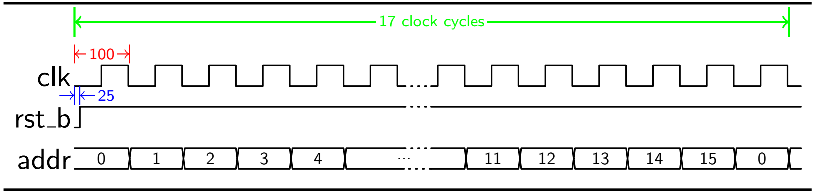


Problem 2

Design a ROM unit for storing the 16 CORDIC *arctan* constants, with the following interface

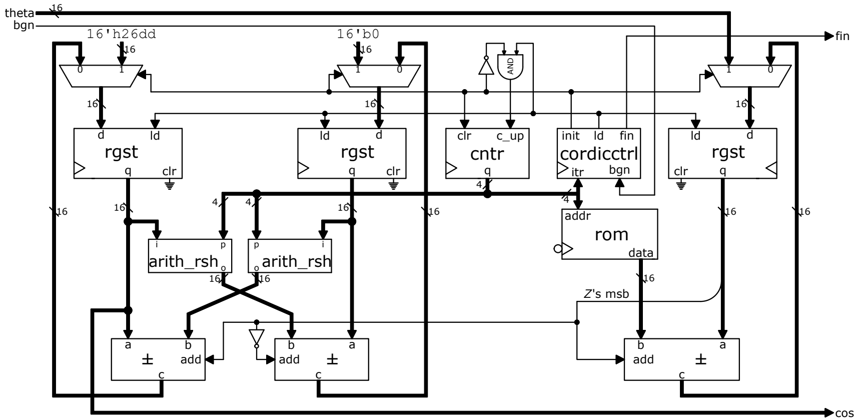
```
module rom #(parameter aw=10, dw=32, file="rom_file.txt")(
  input clk,rst_b,
  input [aw-1:0] addr,
  output reg [dw-1:0] data
);
```

Construct a testbench generating the inputs as in the timing diagram bellow:



Problem 3

Design the *cordic* unit, given its architecture:



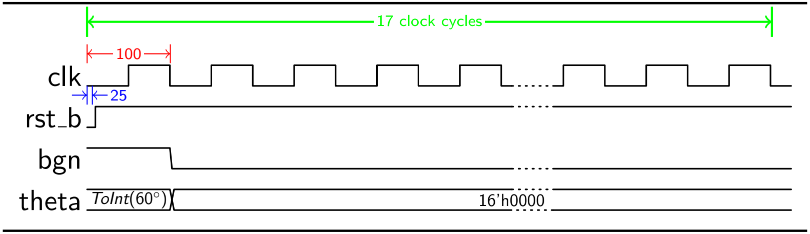
The interface of the module will be:

```verilog
module cordic(
  input clk, rst_b, bgn,
  input [15:0] theta,
  output [15:0] cos
  output fin
);
```

Problem 3 (contd.)

Implement the multiplexers, the arithmetic right shifters and the adder/subtracter structures as Verilog functions.

Build a testbench for approximating the value of $\cos 60^\circ$.



Define a Verilog function *ToRad()* taking as input an angle expressed in degrees and converting it into radians, both input and outputs being of *real* type. Define a Verilog function *ToInt()* taking a real value and converting it into a 16-bit integer according to the format presented in the material. Define a Verilog function *ToReal()* as the inverse operation of *ToInt()*.