ECE 545 Fall 2014
Final Exam

Problem 1 [50 points]

Develop an ASM chart for the key scheduling unit of the RC6 cipher, described below using its

A. **pseudocode**
B. **interface diagram and table of input/output ports**
C. **block diagram**
D. **interface with the division into the datapath and controller.**

Please use ACTIONS (such as S[0] = P32, j=j+1 mod 4, etc.) inside of your states and conditional output boxes, and CONDITIONS (such as i=2r+3) inside of your decision boxes.

Then, redraw your ASM chart, and replace ACTIONS and CONDITIONS by names or expressions involving the corresponding outputs and inputs of the controller.

A. Pseudocode

The key scheduling unit of the RC6 cipher is a circuit that takes a 128-bit key K, and converts it to 2r+4 round keys S[i], i=0..2r+3, stored in the internal memory S[i].

The RC6 key scheduling unit is defined using the given below pseudocode.
The input key K consists of four words K[0], K[1], K[2], and K[3], each of the size of 32 bits.
These words are first written to the internal memory L[j], j=0..3, using the control signal write_key (active high) and the key input Kin. This process is described using the first `for-loop` of the pseudocode. Afterwards, the remaining part of the pseudocode is executed by the controller.
`P32` and `Q32` are 32-bit constants. `r` is a parameter of RC6, with the default value of 12.

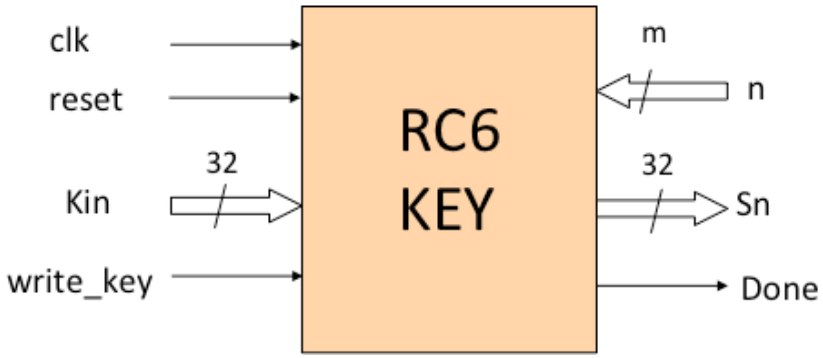
```
for j = 0 to 3 do
    while (not write_key)
          do_nothing
    end while
    L[j] = Kin
end for

S[0] = P32
for i=1 to 2r+3 do
    S[i] = S[i-1] + Q32
end for

i = j = 0
A = B = 0
for k=1 to  3·(2r+4) do
       A = S[i] = (S[i] + A + B) <<< 3
       B = L[j] = (L[j] + A + B) <<< (A+B)
       i = (i+1) mod (2r+4)
       j = (j+1) mod 4
end for
```

Notation:
A, B = 32-bit variables
+ = unsigned addition mod 2^{32}
X <<< Y = rotation of the variable X by a number of positions given by the current value of the variable Y

The circuit includes FPGA-type memories with the following inputs:
ADDR, DIN, CLK, WE, and the output DOUT.
These memories output data in the same clock cycle in which a new address is applied.
Writing to memory takes effect at the rising edge of the clock when WE = 1.

B. Interface and table of input/output ports

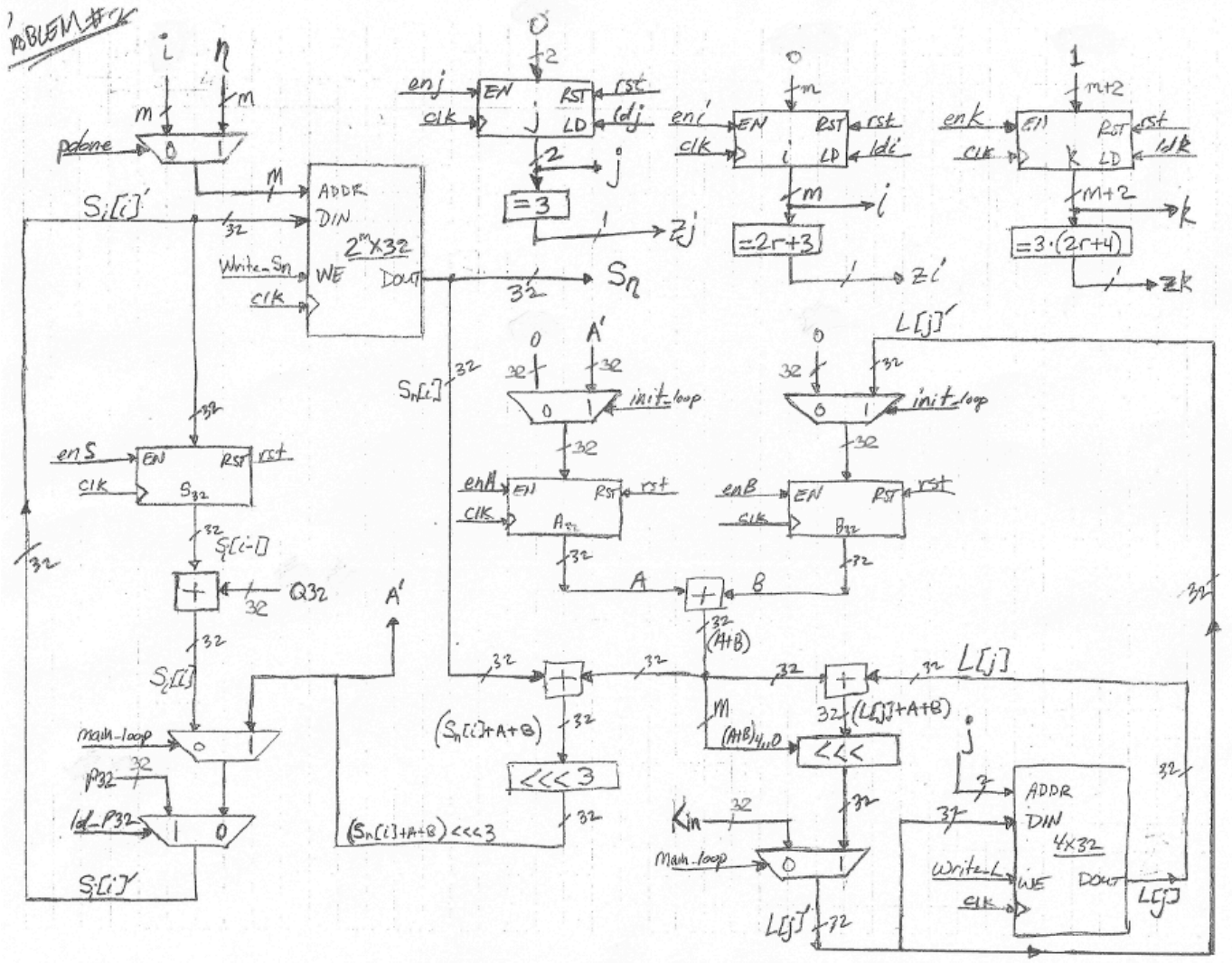
Assume the following interface to your circuit:



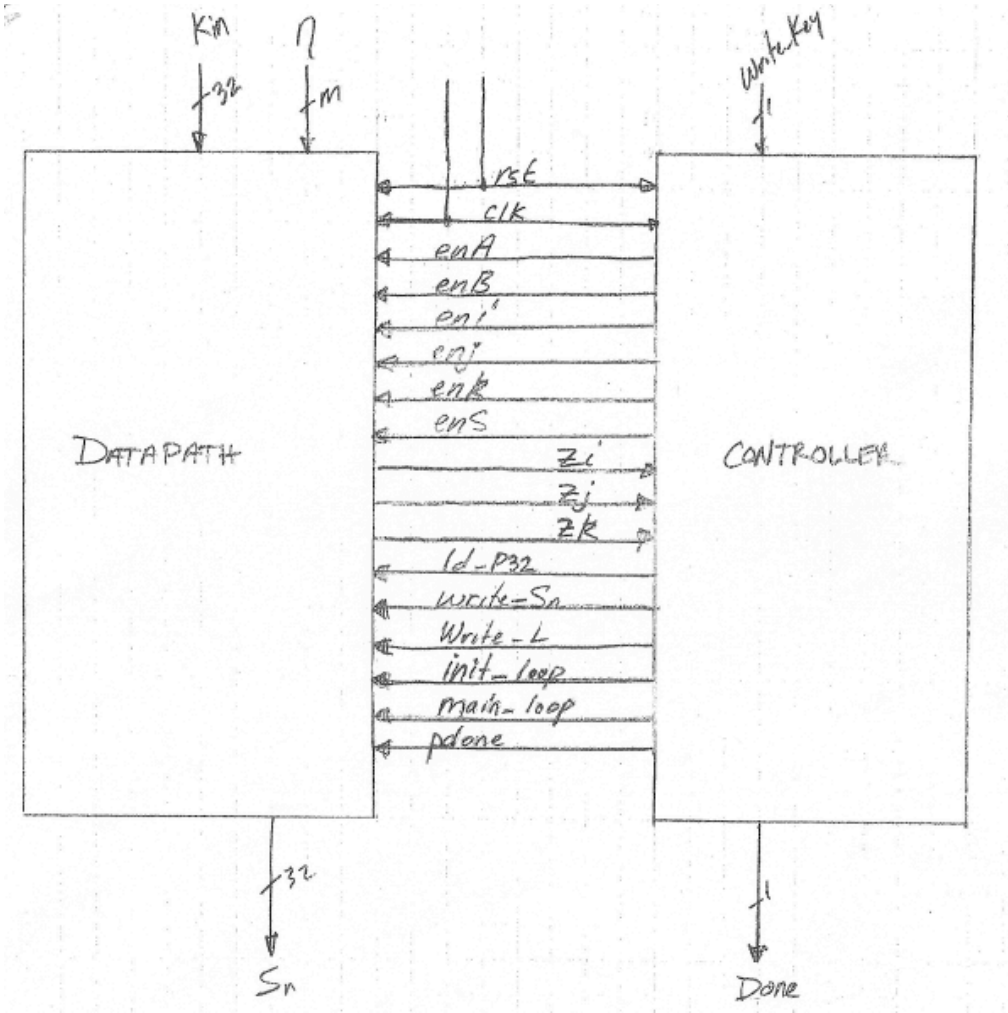
Port	Width	Meaning
clk	1	System clock.
reset	1	System reset – clears internal registers.
write_key	1	Synchronous write control signal for the key input, Kin. Active high.
Kin	32	The key input through which subsequent words of the key K[0]..K[3] are loaded to the circuit.
n	*m*	Index n of the round key S[n].
Sn	32	Value of the round key S[n] corresponding to the index provided through the input n.
Done	1	Asserted when the computations are completed.

m is a size of index n. It is a minimum integer, such that $2^m - 1 \geq 2r+3$.

C. Block diagram



D. Interface with the division into the datapath and controller



Problem 2 [15 points]

Perform timing analysis of the circuit from Problem 1, by answering the following questions:

A. Derive a formula for the

Execution time of this circuit as a function of the parameter r, expressed in clock cycles. Please make sure that this formula is consistent with the operation of your controller!

B. Calculate the Execution Time for r=12 and f_{CLK}=200 MHz.

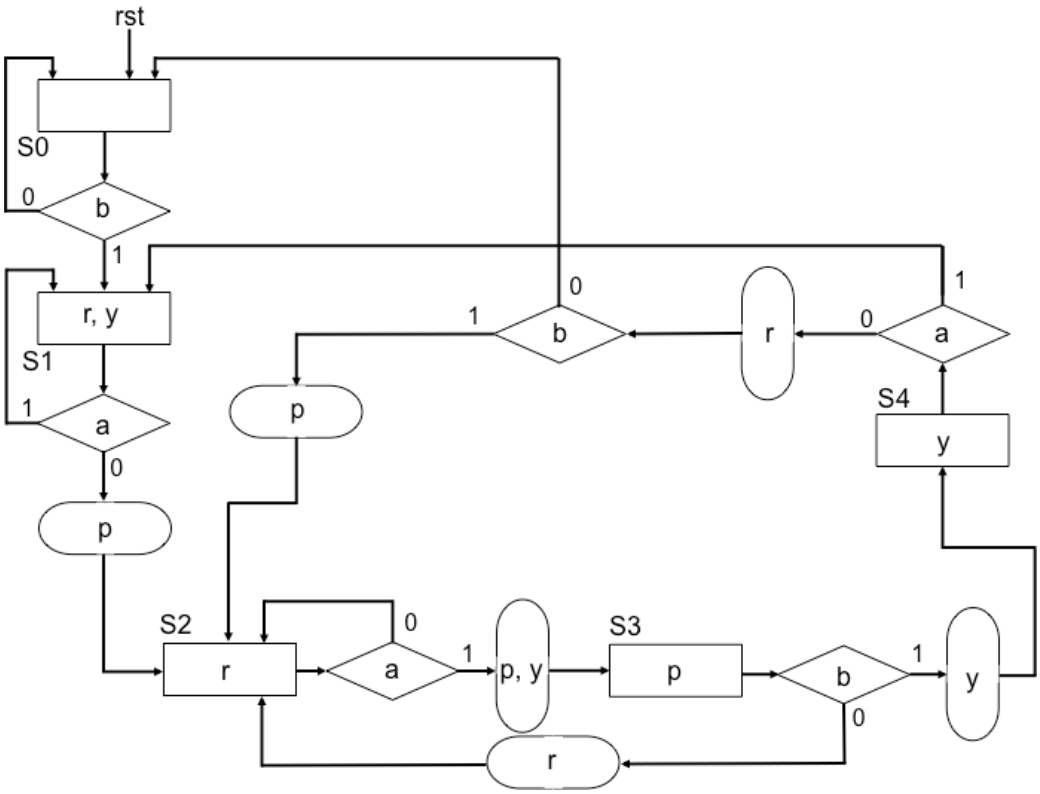
C. Identify the most likely critical path in this circuit, and mark it in your block diagram (in the answer sheet).

D. Express the minimum clock period in terms of delays (and possibly other timing parameters) of all components forming the critical path. Use notation, such as d_{MUX2} – delay of a multiplexer with two inputs, d_{ADD32} – delay of an adder with 32-bit inputs, t_{REG_setup} – setup time of a register, t_{RAM_setup} – setup time of RAM, etc.

Problem 3 [20 points]

Assuming the controller, described using the given below ASM chart:

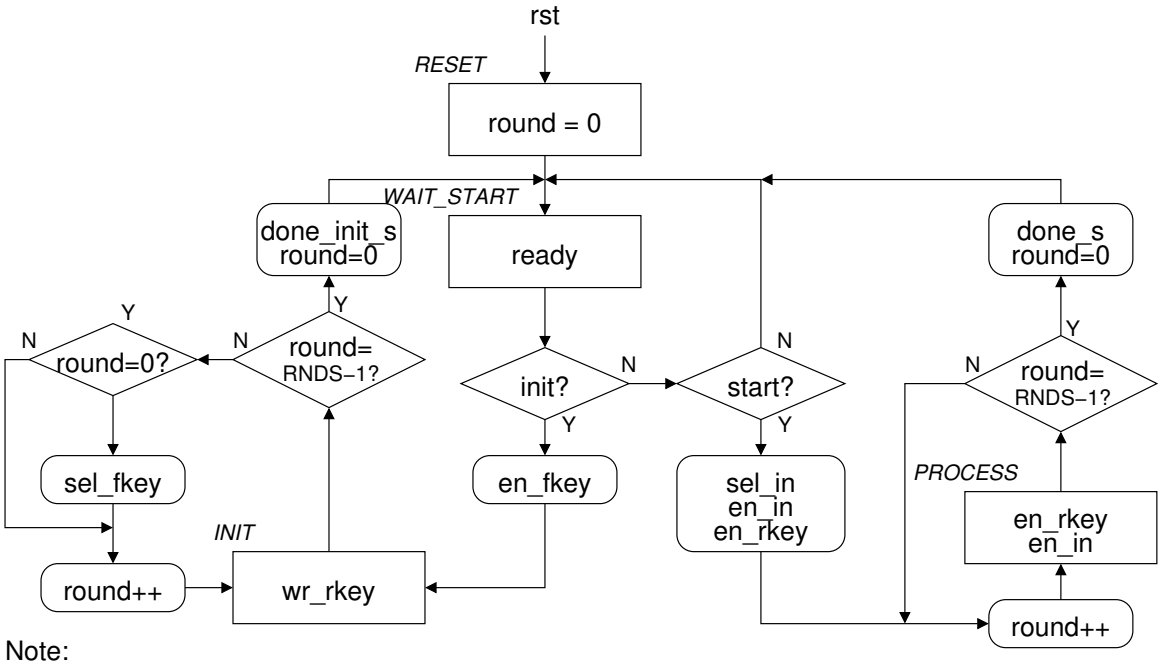
- supplement timing waveforms provided in the answer sheet with the values of the **state S**, and the values of the **outputs p**, **r**, and **y**
- write the VHDL **dataflow** code **for the output function (only)**, calculating **p**, **r**, and **y**.



ASM chart of the controller.

Problem 4 [15 points]

Fill in the blanks in the code of the AES_Enc Controller, **provided in the answer sheet**. **Do not write this code from scratch**! The ASM chart of this Controller is shown below.



Note:
Output of "done" and "done_init" signals are registered.
RNDS = G ROUNDS