CRO Project Details

April 28, 2005

Project Title: **Mini CRO**

Submitted by

- Rahul Garg 2003CS10183
- Varun Gulshan 2003CS10191

Cycle 3
Intructor Name: Vikas Upmanue

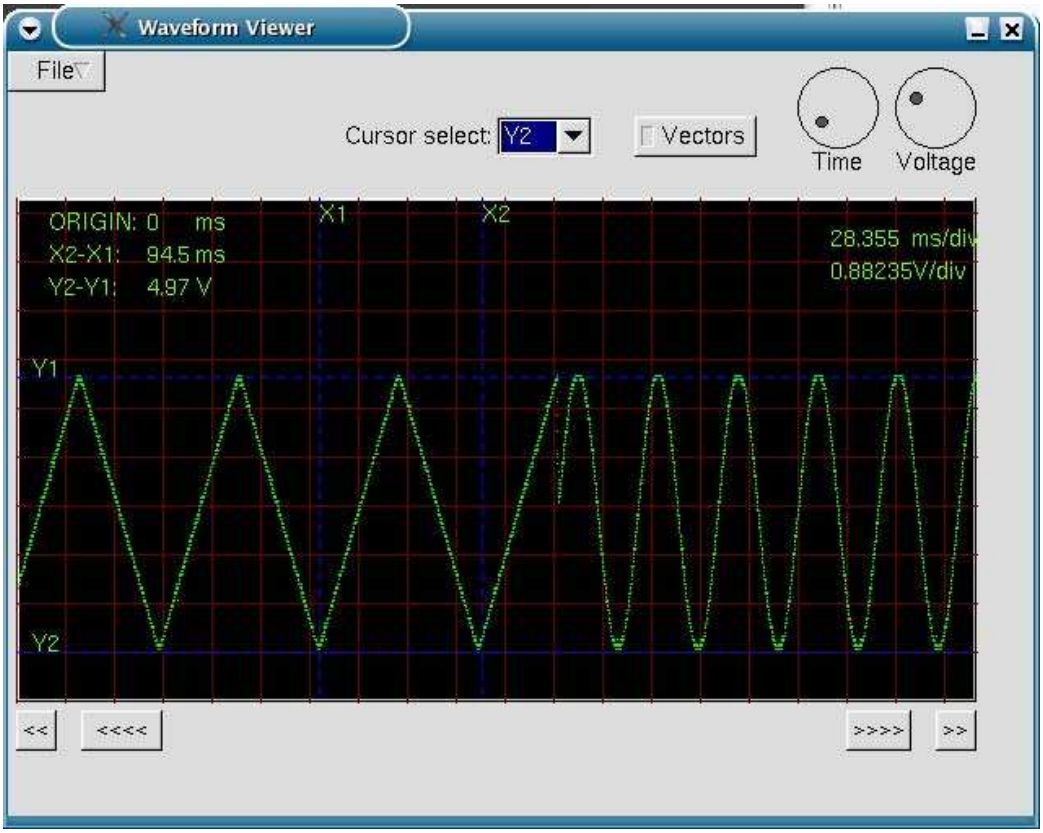
1 Specifications

The mini CRO developed has the following main components:

- An ADC to sample analog data at a maximium of 0.66Mhz
- A 32Kb RAM to store the samples.
- Communicating the information to the computer and plotting the waveform.

The CRO works offline. The user tells the CRO to start sampling through an input. The CRO then samples the data at some fixed frequency(can me made max upto 0.5 Mhz) and stores the samples in the ram. Then the user has to push another button to send this data to the computer(at 38400 bits/sec)(takes about 6 seconds) where a program running receives this data and stores it in a file. This file can then be viewed using the waveform plotter developed.

1.1 Software specifications for the waveform plotter



1. Menubar
   (a) File
      - Open: To open an existing plot
2. Voltage and Time Controls
   (a) A dial to set the voltage scale (i.e volts/div)
   (b) A dial to set the time scale(i.e sec/div)
3. Cursor Controls
   (a) A dropdown menu to select cursor.
   (b) Facility to move the cursors around.
4. Display Screen

(a) Displays the plot of the currently acquired waveform

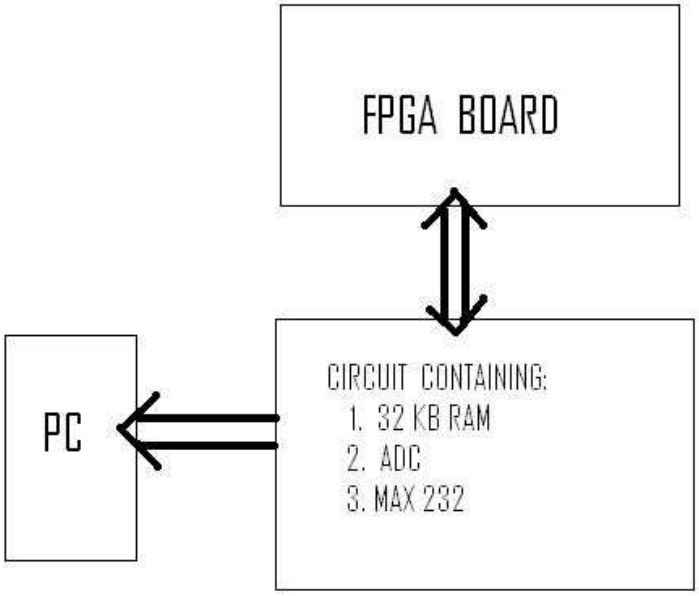
(b) Can be scrolled along X axis.

(c) Has an interactive cursor display where the cursors can be moved around.

(d) Displays information regarding cursors,time scale,voltage scale.

1.2 Hardware specifications

The layout of the circuit is as follows:



The controls and outputs provided to the user are:

1. Sample button: Push the button up to down position to take another sample.The indicator goes off when the sample has been done.

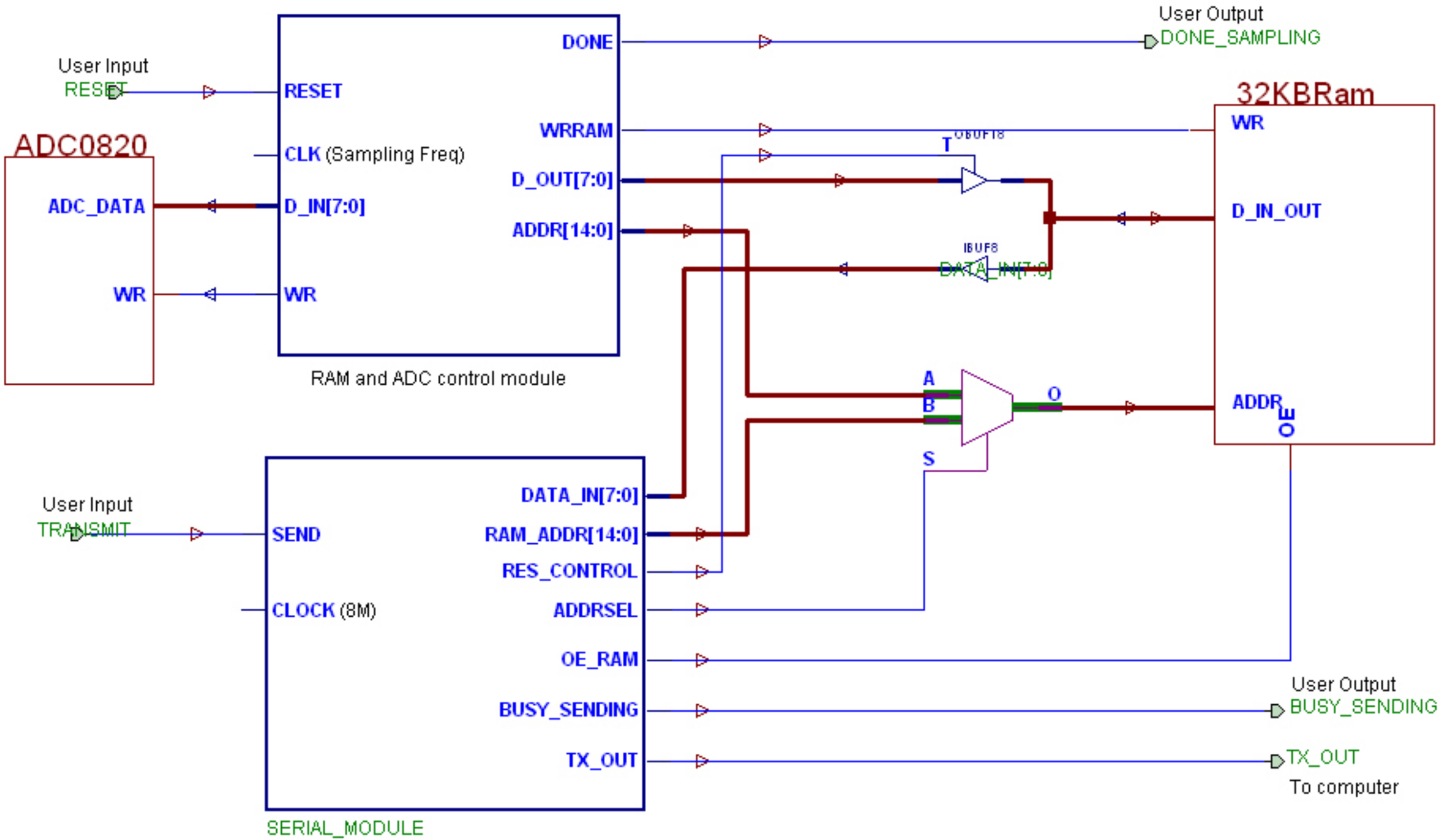
2. Send Button: Push the button from down to up position to send the data to the copmuter.The indicator goes off while the data is being sent to the computer, becomes high once again when the data sending is complete.

3. Sampling indicator led.

4. Sending indicator led.

2 Design Description

The top level schematic is shown on the following page:


User Input
RESET
ADC0820
ADC_DATA
WR
RESET
CLK (Sampling Freq)
D_IN[7:0]
WR
DONE
WRRAM
D_OUT[7:0]
ADDR[14:0]
RAM and ADC control module
User Output
DONE_SAMPLING
32KBRam
WR
D_IN_OUT
ADDR
OBUFT8
T
IBUF8
DATA_IN[7:0]
A
B
O
S
User Input
TRANSMIT
SEND
CLOCK (8M)
DATA_IN[7:0]
RAM_ADDR[14:0]
RES_CONTROL
ADDRSEL
OE_RAM
BUSY_SENDING
TX_OUT
SERIAL_MODULE
User Output
BUSY_SENDING
TX_OUT
To computer

1 DESCRIPTION OF TOP LEVEL SCHEMATIC

The main modules in top level schematic are as follows:

1.1 RAM and ADC CONTROL MODULE

This module communicates with ADC,takes data from it,and writes to ram.

1.1.1 INPUT SIGNALS

1. RESET:This is used to sample new data.
2. CLK:Clock signal,sampling frequency is $\frac{1}{16}$ of the clock frequency.
3. D_IN:Input data from ADC.

1.1.2 OUTPUT SIGNALS

1. WR:write signal to ADC for proper sampling.
2. WRRAM:write signal to RAM for proper writing.
3. D_OUT:Data to be written in RAM.
4. ADDR:Address (15 bit) for RAM.
5. DONE:Goes high,when sampling is complete.

1.2 SERIAL MODULE

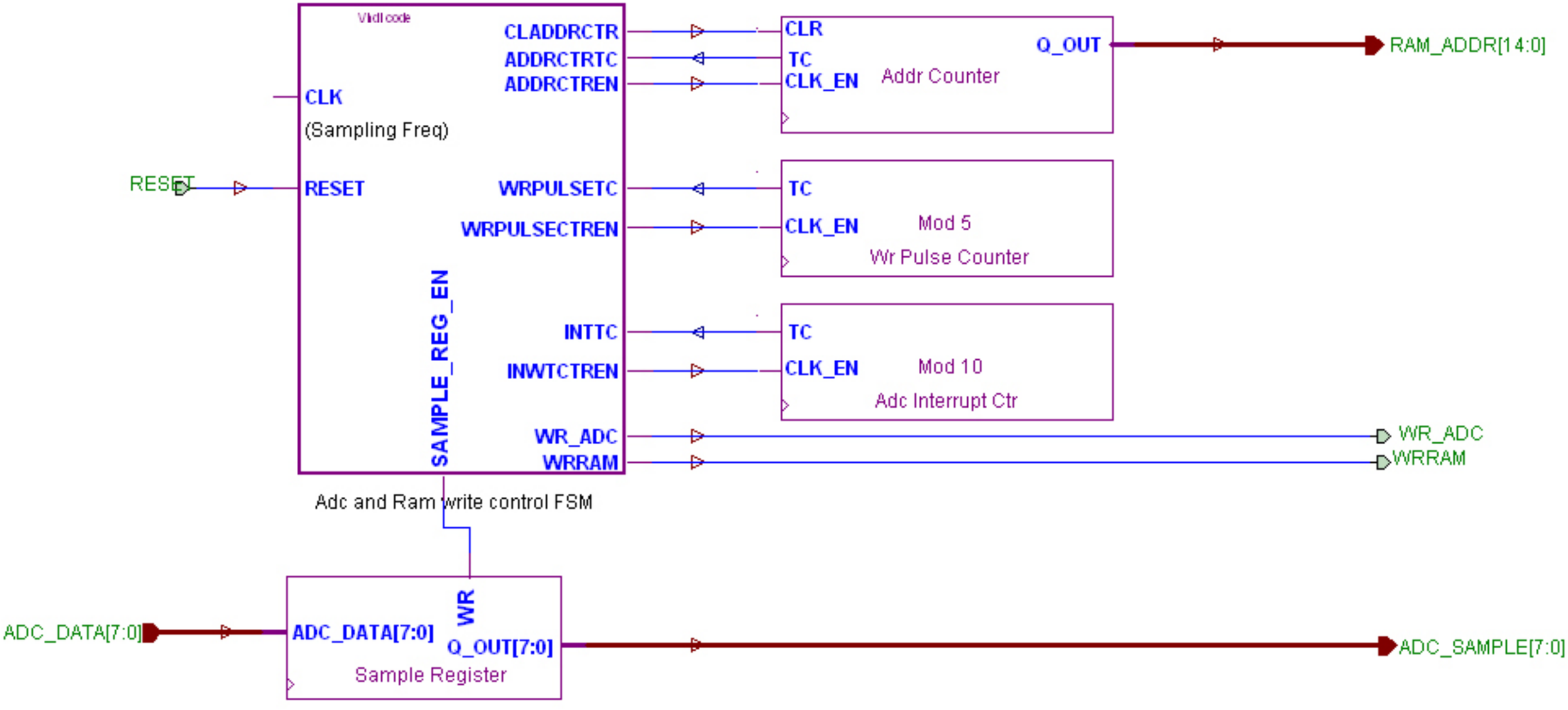
This module sends the data stored in RAM to PC through serial port.

1.2.1 INPUT SIGNALS

1. SEND:User input to start sending data to the PC.
2. CLOCK:8 Mhz clock.
3. DATA_IN:Input data from RAM.

1.2.2 OUTPUT SIGNALS

1. RAM_ADDR:15 bit address for the RAM.
2. RES_CONTROL:resolution control used to control bidirection pins of ram data.
3. ADDRSEL:Input to address select mux,to select between address supplied by RAM and ADC MODULE and this module.
4. OE_RAM:Output Enable of RAM.
5. BUSY_SENDING:Is high,when the circuit is busy sending data.
6. TX_OUT:Signal for the TX line of serial port.



Ram writing and ADC Sampling control module

1 DESCRIPTION OF RAM and ADC CONTROL MODULE

The main submodules in it are as folilows:

1.1 ADC and RAM WRITE CONTROL FSM

This FSM basically generates signals to sample data from ADC,write data to the RAM and latching the ADC data onto a register. The signals are described in detail as follows.

1.1.1 INPUT SIGNALS

1. RESET:This is used to sample new data.
2. CLK:Clock signal,sampling frequency is $\frac{1}{16}$ of the clock frequency.
3. ADDRCTRTC:Terminal count of address counter.
4. WRPULSETC:Terminal count of write pulse counter.
5. INTTC:Terminal count of Adc interrupt counter.

1.1.2 OUTPUT SIGNALS

1. CLADDRCTR:clear address counter.
2. ADDRCTREN:enable address counter.
3. WRPULSECTREN:enable write pulse counter.
4. INWTCTREN:enable Adc interrupt counter.
5. WR_ADC:write signal for ADC.
6. WRRAM:Write signal for RAM.
7. SAMPLE_REG_EN:Enable sample register.

1.2 SAMPLE REGISTER

This register stores the data recieved from ADC so that it can be written to RAM during subsequent cycles.

1.3 ADDRESS COUNTER

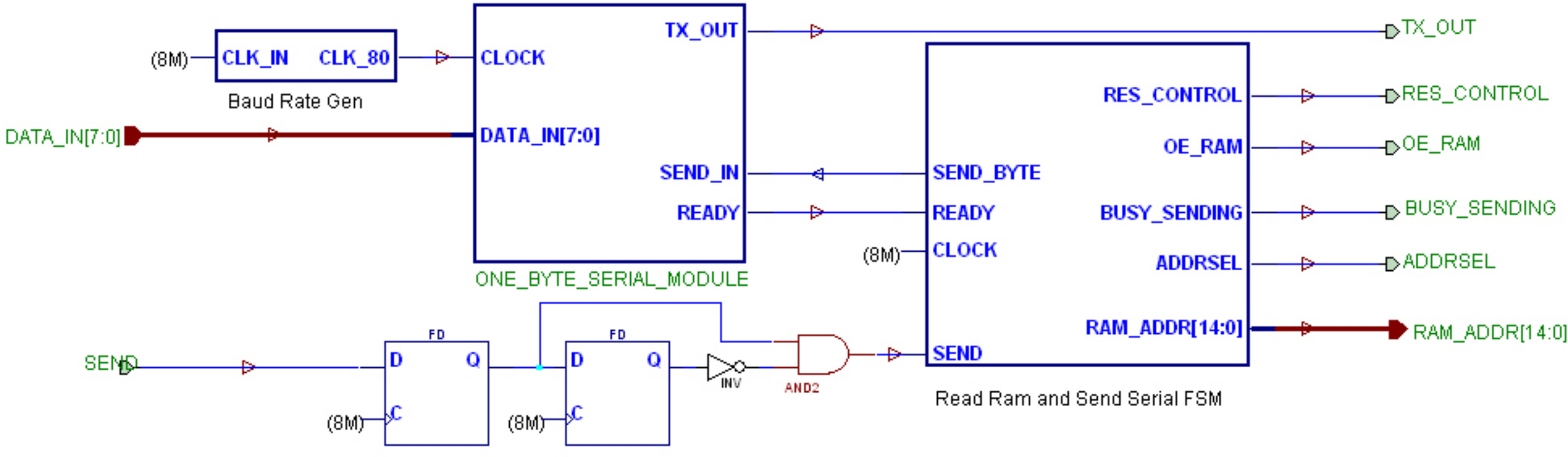
This counter generates 15 bit address for the ram.

1.4 WRITE PULSE COUNTER

A mod 5 counter so that write pulses may be generated at appropriate timing.

1.5 ADC INTERRUPT COUNTER

A mod 10 counter(to provide appropriate time to the ADC for conversion).


(8M)
CLK_IN
CLK_80
Baud Rate Gen
DATA_IN[7:0]
CLOCK
DATA_IN[7:0]
TX_OUT
SEND_IN
READY
ONE_BYTE_SERIAL_MODULE
TX_OUT
RES_CONTROL
OE_RAM
BUSY_SENDING
ADDRSEL
RAM_ADDR[14:0]
SEND_BYTE
READY
(8M)
CLOCK
SEND
Read Ram and Send Serial FSM
RES_CONTROL
OE_RAM
BUSY_SENDING
ADDRSEL
RAM_ADDR[14:0]
SEND
FD
D
Q
C
(8M)
FD
D
Q
C
(8M)
INV
AND2

1 DESCRIPTION OF SERIAL MODULE

The main submodules in it are as follows:

1.1 ONE BYTE SERIAL MODULE

This module sends one byte of data to the PC:

1.1.1 INPUT SIGNALS

1. DATA_IN:One byte of data to be sent.
2. CLK:Clock signal.It should be of appropriate frequency to match the baud rate.
3. SEND_IN:Signal to start sending data to the PC.

1.1.2 OUTPUT SIGNALS

1. TX_OUT:Signal for TX line of serial port.
2. READY:This signal is high if the module is ready to send new data.

1.2 BAUD RATE GENERATER

It adjusts the clock freq to match the baud rate requirements.

1.3 READ RAM AND SEND SERIAL FSM

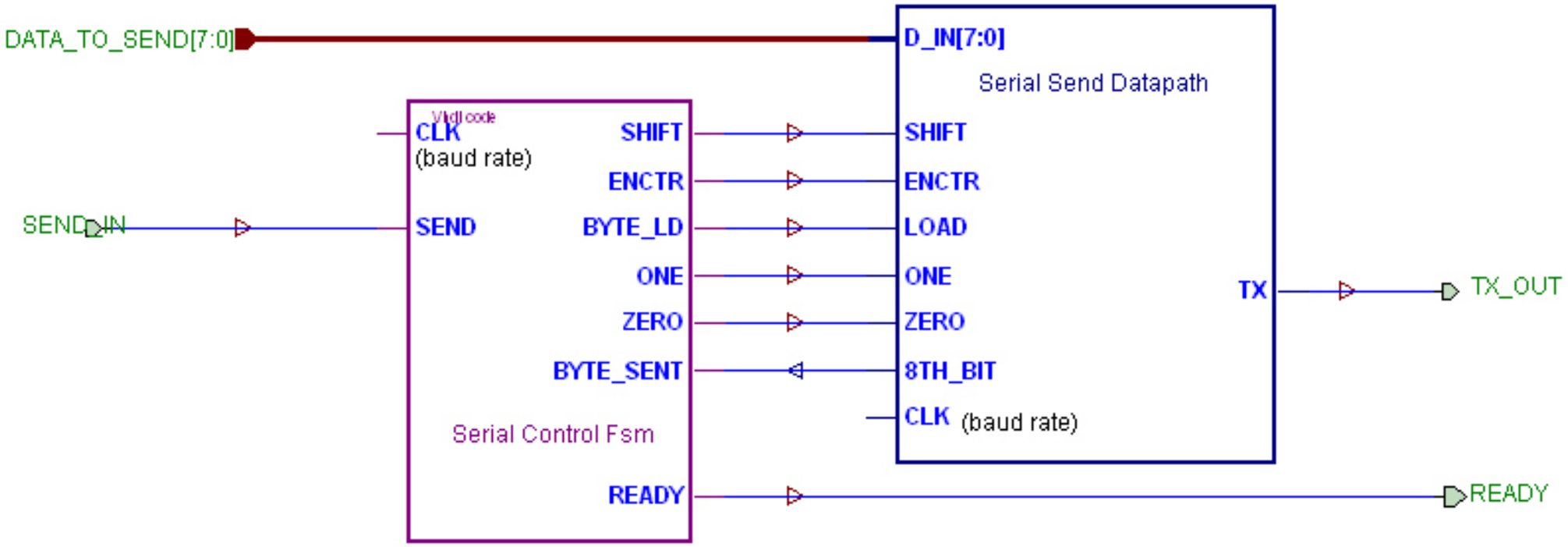
This fsm generates signals to read from ram and communicates with the one byte serial module to send data to the pc.

1.3.1 INPUT SIGNALS

1. READY:Input signal from one byte to serial module to signal whether it is ready to send new data or not.
2. CLOCK:clock signal.
3. SEND:User input to send data from ram(This signal is waveshaped before sending it to FSM).

1.3.2 OUTPUT SIGNALS

1. SEND_BYTE:signal to send for one byte serial module.
2. RES_CONTROL:resolution control signal for data lines of ram.
3. OE_RAM:output enable signal of ram.
4. BUSY_SENDING:Goes high if the module is busy sending data to the PC.
5. ADDRSEL:Signal(to mux) to select address(from ram and adc control module or this module).
6. RAM_ADDR:15 bit address for the ram.



One Byte Serial Sending Module

1 ONE BYTE SERIAL MODULE

The main submodules in it are as follows:

1.1 SERIAL CONTROL FSM

This fsm generates signals to load the data to be sent on to the shift register, shift the register and generates start and stop bits

1.1.1 INPUT SIGNALS

1. SEND:Signal to start sending data to the PC.
2. CLK:Clock signal.It should be of appropriate frequency to match the baud rate.
3. BYTE_SENT:Signal to indicate that the current byte has been sent.

1.1.2 OUTPUT SIGNALS

1. SHIFT:To enable shift register in serial datapath.
2. ENCTR:To enable (mod 8) counter.
3. LOAD:To load shift register.
4. ONE and ZERO:These signals are used in generation of start/stop bit(on TX_OUT).
5. READY:This signal is high if it is not busy sending any data.

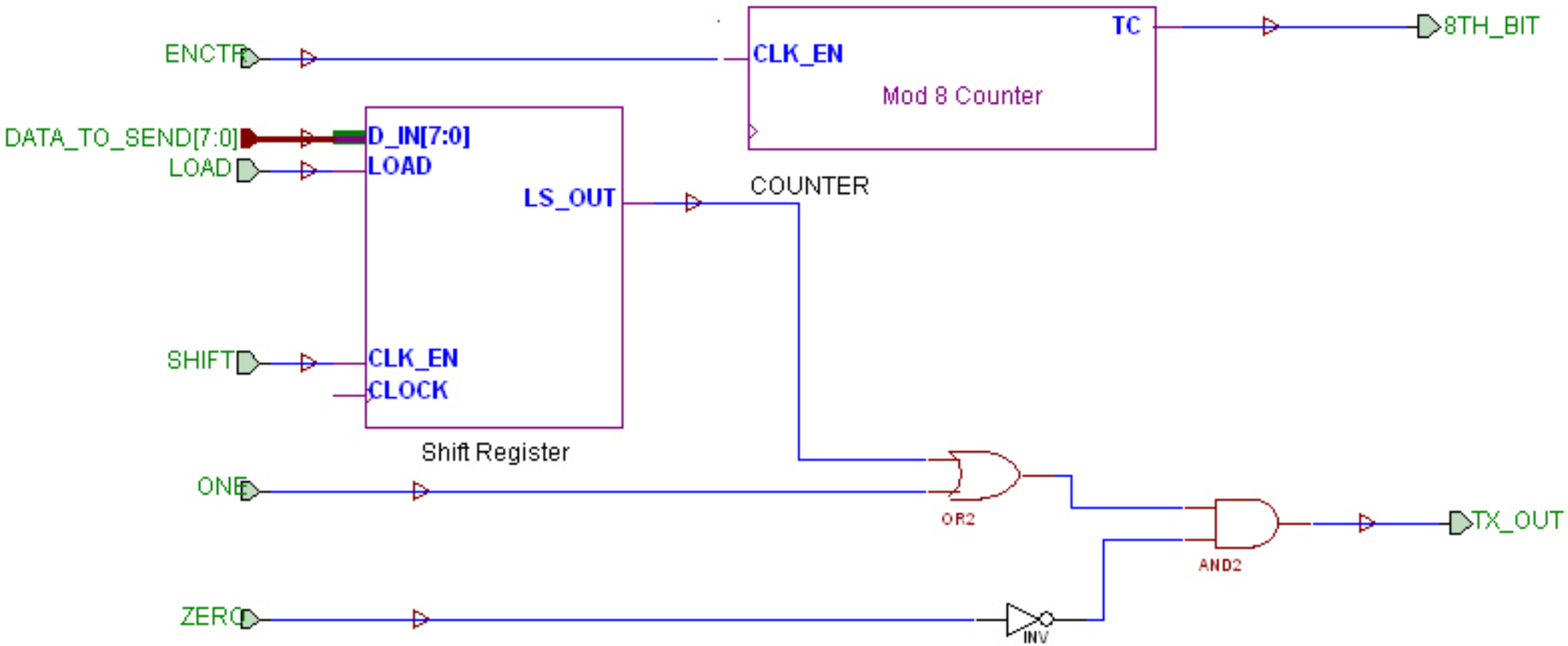
1.2 SERIAL SEND DATAPATH

1.2.1 INPUT SIGNALS

1. SHIFT:Shift signal for shift register.
2. ENCTR:To enable (mod 8) counter.
3. LOAD:To load shift register.
4. ONE and ZERO:These signals are used in generation of start bit(on TX_OUT).
5. CLOCK:clock signal.
6. D_IN:Input data to be sent.

1.2.2 OUTPUT SIGNALS

1. TX_OUT:signal for TX terminal of serial port.
2. 8TH_BIT:signal to indicate that all 8 bits have been sent.



Serial Datapath

1 SERIAL DATAPATH

The main submodules in it are as follows:

1.1 SHIFT REGISTER

This shift register is used to send data serially.

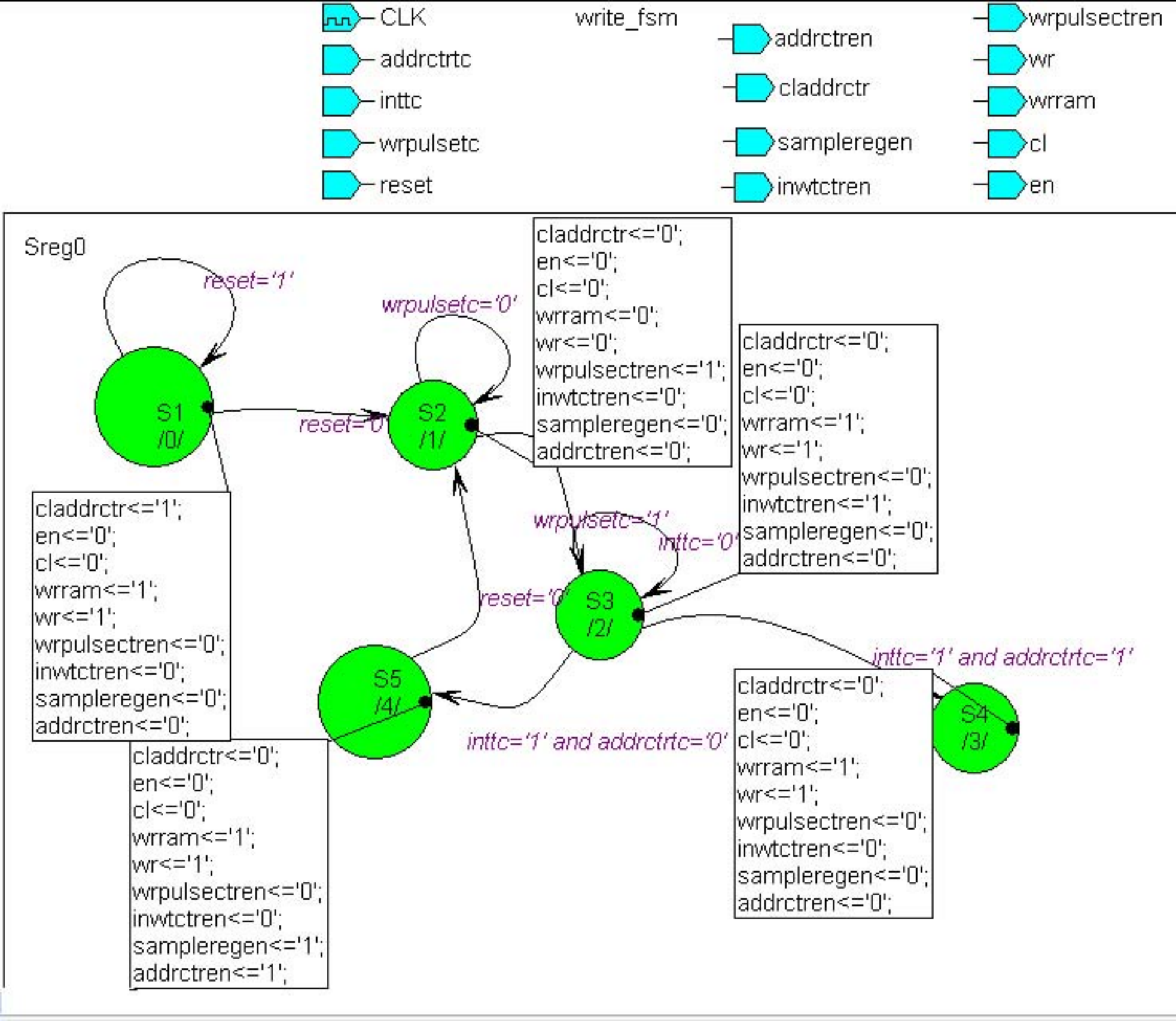
1.2 MOD 8 COUNTER

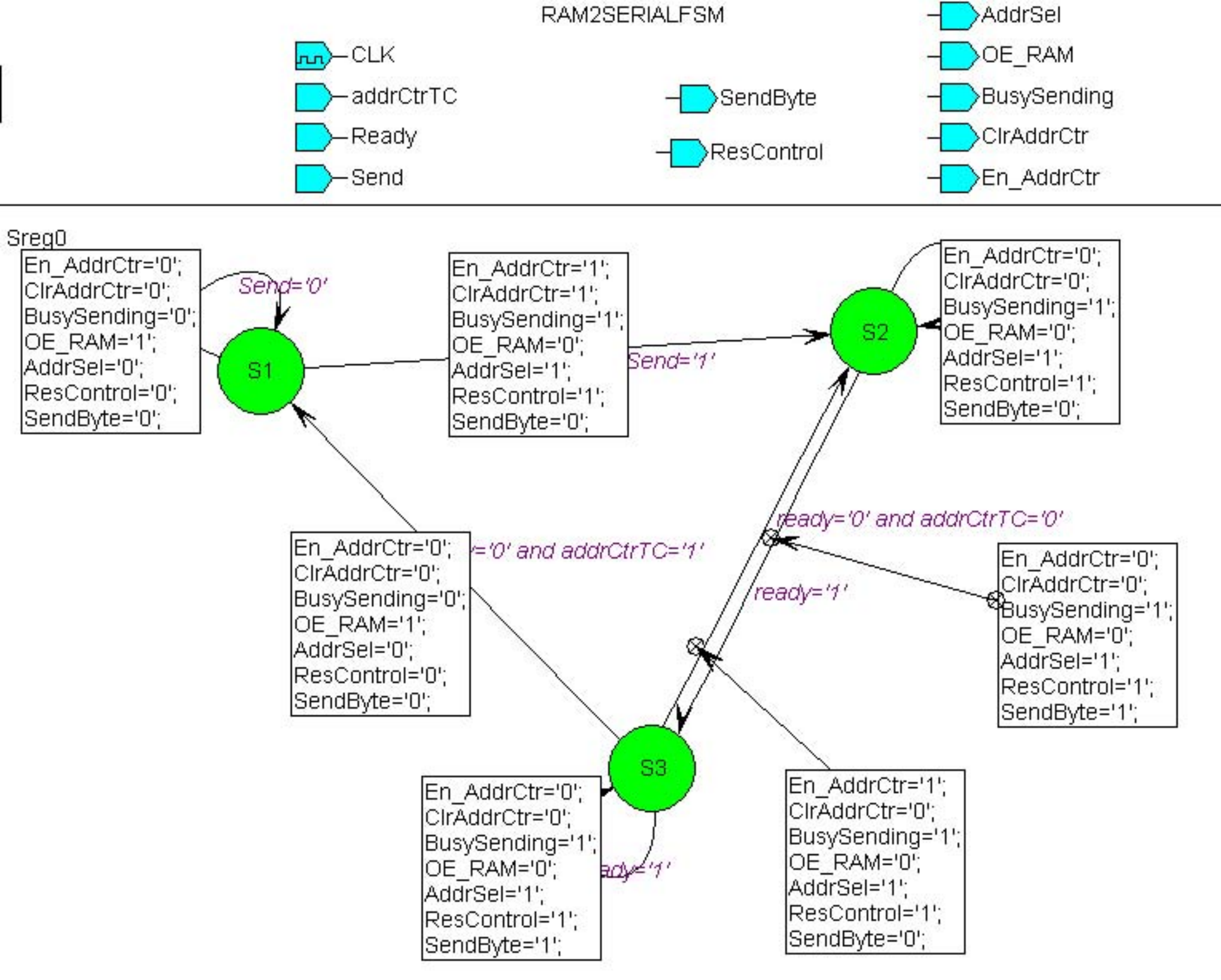
This counter is basically used to count the no. of shifts.

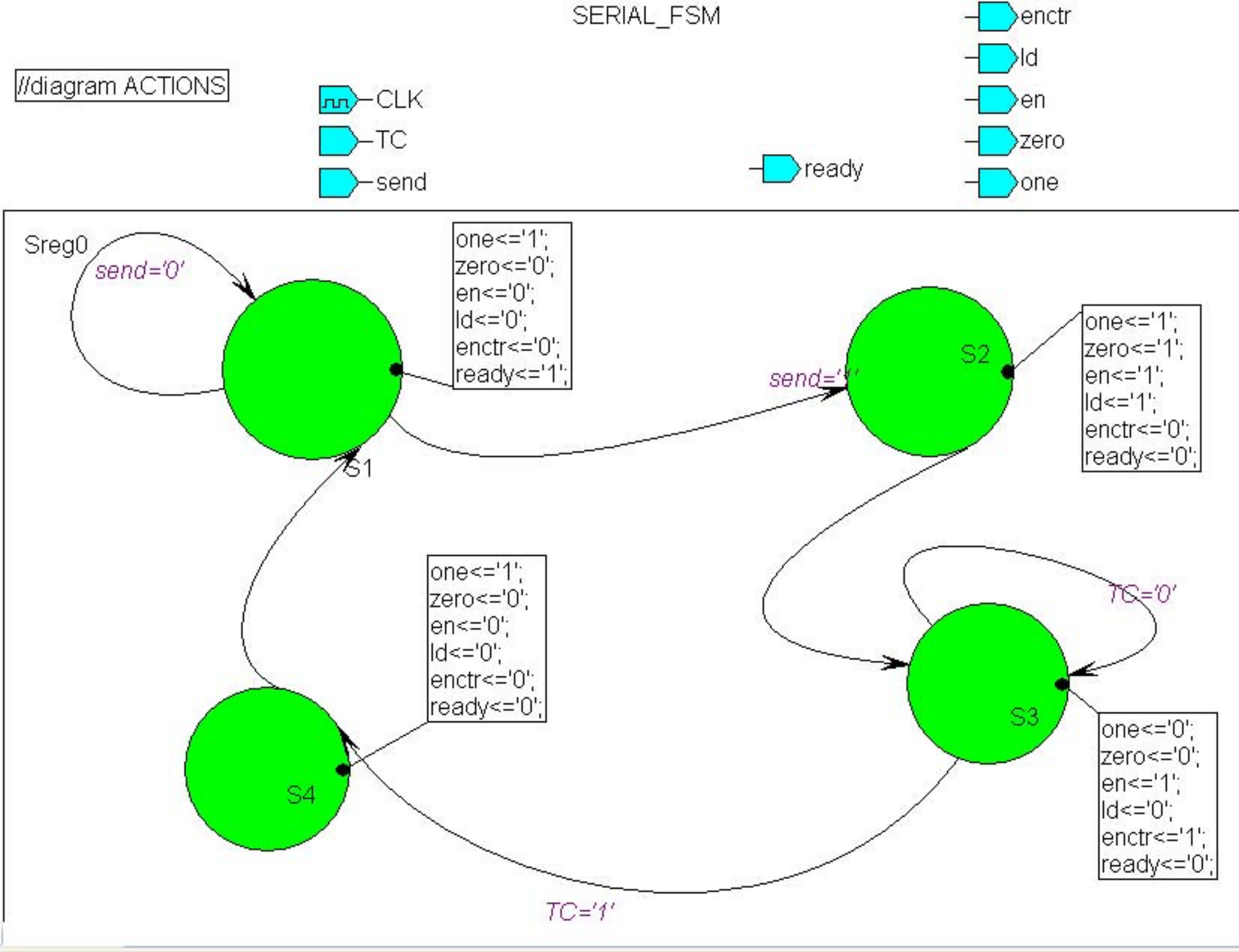
A combination circuit is used to geenrate start and stop bits at appropriate time(using one and zero inputs from FSM).

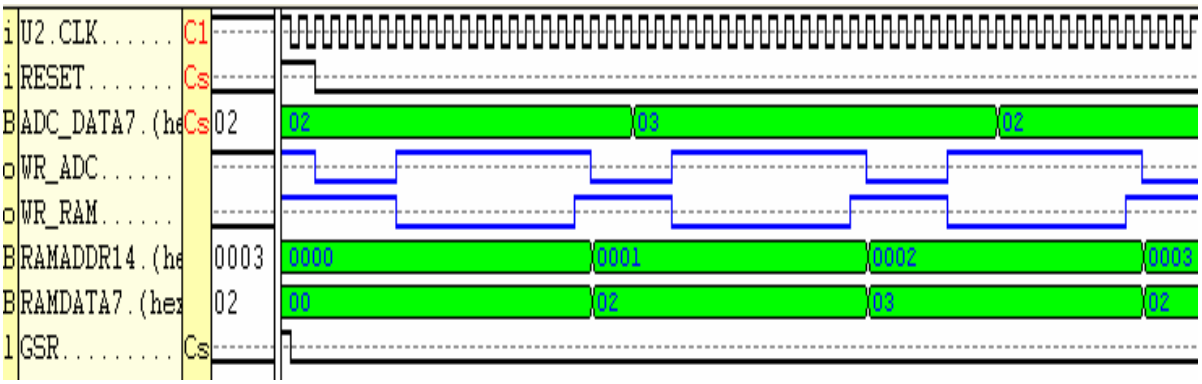
2 State machine diagrams

1. Write FSM: controls sampling and writing of samples onto RAM.
2. RamToSerial FSM: controls reading data from RAM and signalling the serial fsm to send this data.
3. Serial FSM: controls the sending of one byte of information to the PC.


CLK
addrctrtc
inttc
wrpulsetc
reset
write_fsm
addrctren
claddrctr
sampleregen
inwtctren
wrpulsectren
wr
wrram
cl
en
Sreg0
reset='1'
wrpulsetc='0'
reset='0'
wrpulsetc='1'
inttc='0'
reset='0'
inttc='1' and addrctrtc='1'
inttc='1' and addrctrtc='0'
S1
/0/
S2
/1/
S3
/2/
S4
/3/
S5
/4/
claddrctr<='0';
en<='0';
cl<='0';
wrram<='0';
wr<='0';
wrpulsectren<='1';
inwtctren<='0';
sampleregen<='0';
addrctren<='0';
claddrctr<='0';
en<='0';
cl<='0';
wrram<='1';
wr<='1';
wrpulsectren<='0';
inwtctren<='1';
sampleregen<='0';
addrctren<='0';
claddrctr<='1';
en<='0';
cl<='0';
wrram<='1';
wr<='1';
wrpulsectren<='0';
inwtctren<='0';
sampleregen<='0';
addrctren<='0';
claddrctr<='0';
en<='0';
cl<='0';
wrram<='1';
wr<='1';
wrpulsectren<='0';
inwtctren<='0';
sampleregen<='0';
addrctren<='0';
claddrctr<='0';
en<='0';
cl<='0';
wrram<='1';
wr<='1';
wrpulsectren<='0';
inwtctren<='0';
sampleregen<='1';
addrctren<='1';

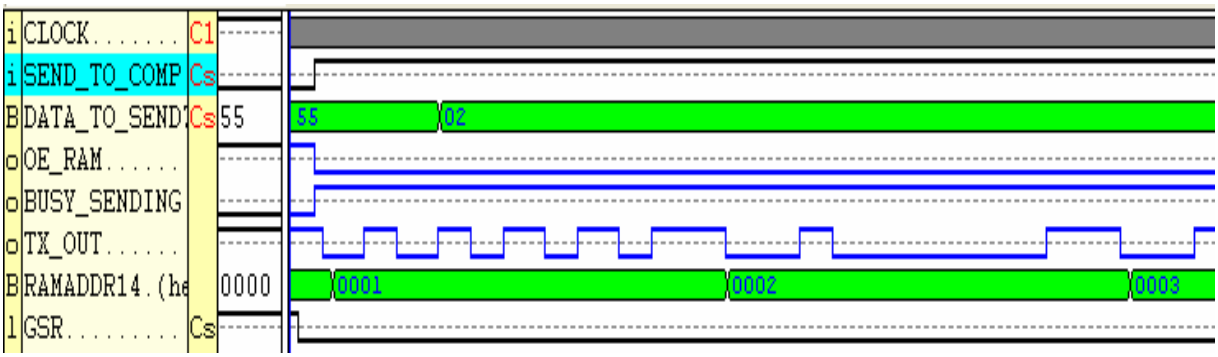

RAM2SERIALFSM
CLK
addrCtrTC
Ready
Send
SendByte
ResControl
AddrSel
OE_RAM
BusySending
ClrAddrCtr
En_AddrCtr
Sreg0
En_AddrCtr='0';
ClrAddrCtr='0';
BusySending='0';
OE_RAM='1';
AddrSel='0';
ResControl='0';
SendByte='0';
Send='0'
S1
En_AddrCtr='1';
ClrAddrCtr='1';
BusySending='1';
OE_RAM='0';
AddrSel='1';
ResControl='1';
SendByte='0';
Send='1'
S2
En_AddrCtr='0';
ClrAddrCtr='0';
BusySending='1';
OE_RAM='0';
AddrSel='1';
ResControl='1';
SendByte='0';
ready='0' and addrCtrTC='0'
En_AddrCtr='0';
ClrAddrCtr='0';
BusySending='0';
OE_RAM='1';
AddrSel='0';
ResControl='0';
SendByte='0';
ready='1'
En_AddrCtr='0';
ClrAddrCtr='0';
BusySending='1';
OE_RAM='0';
AddrSel='1';
ResControl='1';
SendByte='1';
S3
En_AddrCtr='0';
ClrAddrCtr='0';
BusySending='1';
OE_RAM='0';
AddrSel='1';
ResControl='1';
SendByte='1';
En_AddrCtr='1';
ClrAddrCtr='0';
BusySending='1';
OE_RAM='0';
AddrSel='1';
ResControl='1';
SendByte='0';


SERIAL_FSM
//diagram ACTIONS
CLK
TC
send
ready
enctr
ld
en
zero
one
Sreg0
send='0'
one<='1';
zero<='0';
en<='0';
ld<='0';
enctr<='0';
ready<='1';
S1
send='1'
S2
one<='1';
zero<='1';
en<='1';
ld<='1';
enctr<='0';
ready<='0';
TC='0'
S3
one<='0';
zero<='0';
en<='1';
ld<='0';
enctr<='1';
ready<='0';
S4
one<='1';
zero<='0';
en<='0';
ld<='0';
enctr<='0';
ready<='0';
TC='1'



SIMULATION OF WRITE MODULE

RESET: Sampling starts when reset goes low.
ADC_DATA [7-0]: Sample from ADC.
WR_ADC: Write pulse generated by the state machine for ADC.
WR_RAM: Write pulse generated by the state machine for writing RAM. Ram address and data are stable when the write pulse goes low.
RAMADDR [14-0]: Address (15-bit) for RAM.
RAMDATA [7-0]: Data to be written into RAM.



SIMULATION OF SERIAL MODULE

CLOCK: Clock signal, which is converted internally to match the baud rate.
SEND_TO_COMP: User input to start sending data from RAM to PC.
DATA_TO_SEND: Data read from RAM (to be sent on serial port).
OE_RAM: Output enable signal of RAM.
BUSY_SENDING: Goes high, if the module is busy sending data.
TX_OUT: Signal for TX line of serial port.
RAMADDR [14-0]: Address (15-bit) for RAM.

Implementation report summary

Device utilization summary:

```
Number of External IOBs           41 out of 160    67%
Flops:                         0
Latches:                       0

Number of CLBs                    62 out of 400    15%
Total CLB Flops:                  75 out of 800     9%
4 input LUTs:                     105 out of 800    13%
3 input LUTs:                     14 out of 400     3%
```

Timing Report

```
=========================================================================
Timing constraint: Default period analysis

 1941 items analyzed, 0 timing errors detected.
 Minimum period is  17.948ns.
 Maximum delay is  31.422ns.
-------------------------------------------------------------------------

=========================================================================
Timing constraint: Default net enumeration

 156 items analyzed, 0 timing errors detected.
 Maximum net delay is  13.334ns.
-------------------------------------------------------------------------


All constraints were met.
```

```
Data Sheet report:
-----------------
All values displayed in nanoseconds (ns)
```

Setup/Hold to clock CLOCK

Source Pad	Setup to clk (edge)	Hold to clk (edge)
ADC_IN0	-7.025(R)	8.025(R)
ADC_IN1	-6.262(R)	8.362(R)
ADC_IN2	-10.226(R)	11.226(R)
ADC_IN3	-7.310(R)	9.410(R)
ADC_IN4	-8.710(R)	9.710(R)
ADC_IN5	-7.171(R)	9.271(R)
ADC_IN6	-10.611(R)	11.611(R)
ADC_IN7	-7.809(R)	9.909(R)
RESET	0.819(R)	5.620(R)
SEND_TO_COMP	-4.560(R)	5.560(R)

Clock CLOCK to Pad

Destination Pad	clk (edge) to PAD
ADD0	35.418(R)
ADD1	33.589(R)
ADD10	29.921(R)
ADD11	35.798(R)
ADD12	36.061(R)
ADD13	32.445(R)
ADD14	33.661(R)
ADD2	34.188(R)
ADD3	33.147(R)
ADD4	34.031(R)
ADD5	35.231(R)
ADD6	37.648(R)
ADD7	34.739(R)
ADD8	37.179(R)
ADD9	34.912(R)
BUSY_SENDING	31.107(R)
DONE_SAMPLING	26.401(R)
OE_RAM	27.731(R)
RAM_DATA<0>	33.397(R)
RAM_DATA<1>	34.965(R)
RAM_DATA<2>	33.397(R)
RAM_DATA<3>	33.291(R)
RAM_DATA<4>	32.073(R)
RAM_DATA<5>	32.073(R)
RAM_DATA<6>	32.073(R)
RAM_DATA<7>	32.073(R)
TEST	28.809(R)
WR_ADC	28.347(R)
WR_RAM	29.711(R)

Limitations and possible improvements:

- RAM size is small, hence the product (sample length * sampling frequency) is bounded.
- Currently only positive voltages(0-5V)are sampled.
- Trigger not supported because of unavailability of pins on the board.
- CRO is offline because of slow speed of serial port (3840 bytes/sec) as compared to the sampling rate(31.2 KB/sec – 500 KB/sec). It can be made online by using faster communication like USB.