EE417 Advanced Digital Logic with Verilog HDL Name: Ron Kalin / Lamin Jammeh Date: 07/03/24

Lesson 8 **Universal Asynchronous Receiver Transmitter**

One of the widely used communication protocols is the UART (Universal Asynchronous Receiver Transmitter). The book pages 388 - 399 have detailed information about UART receiver. Translate the code into a block diagram that includes the DataPath and the Controller modules. Based on the Controller code sketch the state graph of the FSM, and based on the DataPath design, sketch the internal registers, the dataflow through them and what control signals they wait for to complete a data transmission or state transition.

Main objective: understand how the UART receiver works.

Block Diagram

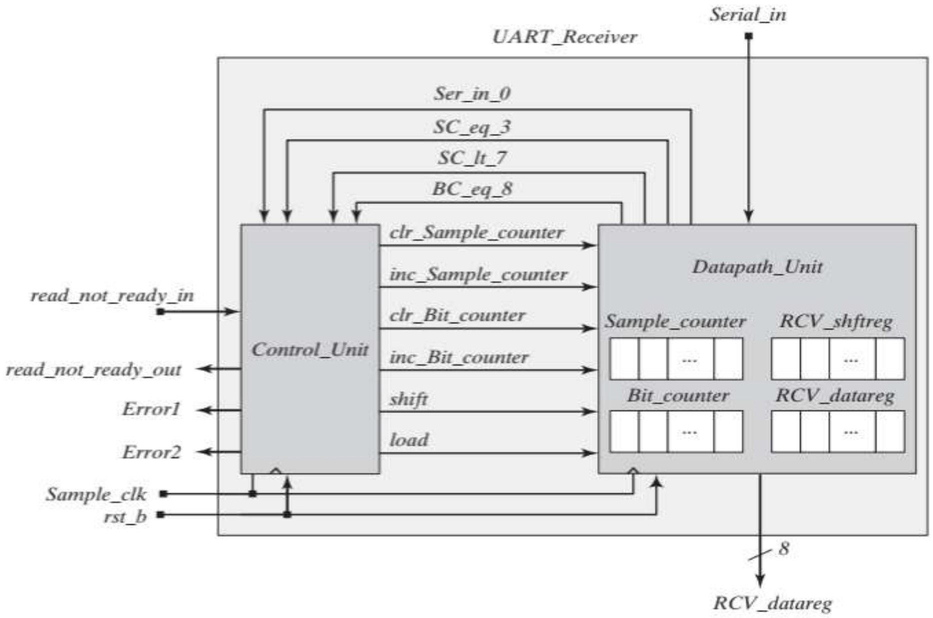


FIGURE 7-25 Block diagram of *UART_receiver*, including the interface signals between the control unit and the datapath unit.

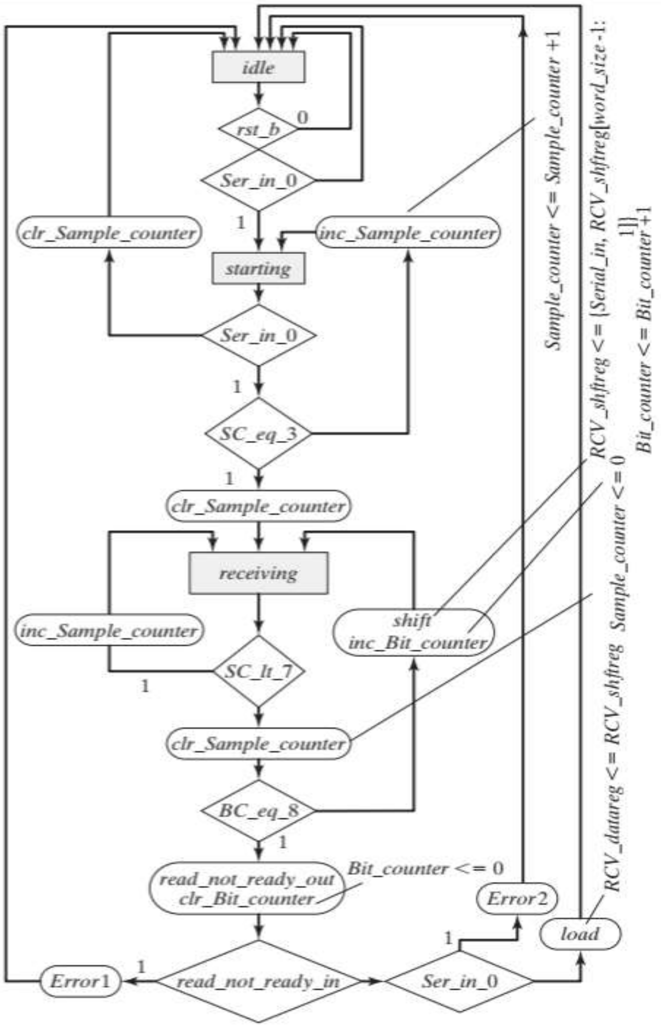
The state machine has the following primary (external) inputs and status inputs:

read_not_ready_in	signals that the host is not ready to receive data
Ser_in_0	asserts while *Serial_in* is 0
SC_eq_3	asserts while *Sample_counter* = 3
SC_lt_7	asserts while *Sample_counter* < 7
BC_eq_8	asserts while *Bit_counter* = 8
Sample_counter	counts the samples of a bit
Bit_counter	counts the bits that have been sampled

The state machine produces the following outputs:

read_not_ready_out	signals that the receiver has received 8 bits
clr_Sample_counter	clears *Sample_counter*
inc_Sample_counter	increments *Sample_counter*
clr_Bit_counter	clears *Bit_counter*
inc_Bit_counter	increments *Bit_counter*
shift	causes *RCV_shftreg* to shift towards the LSB
load	causes *RCV_shftreg* to transfer data to *RCV_datareg*
Error1	asserts if host is not ready to receive data after last bit has been sampled
Error2	asserts if the stop-bit is missing

ASMD Flow chart -Controller Rx

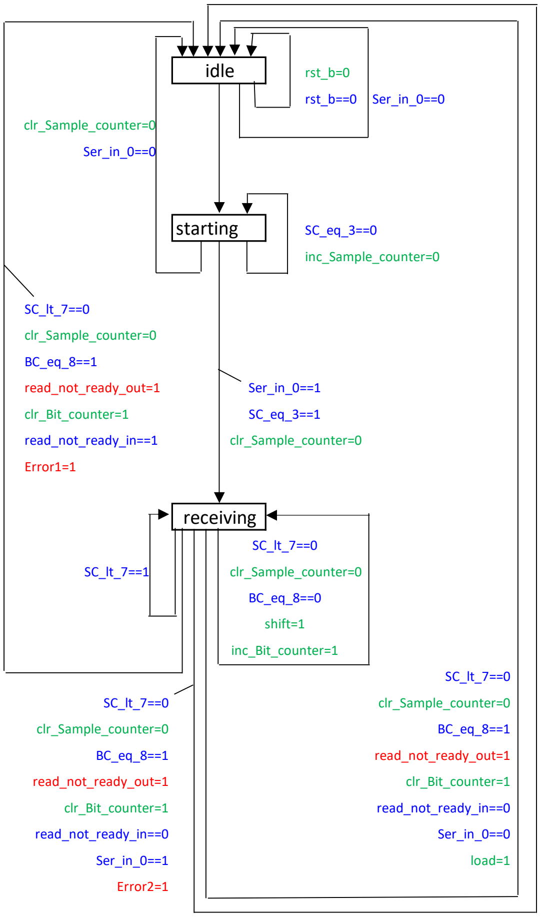


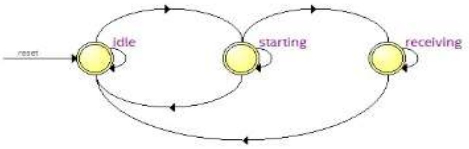
Note: *Ser_in_0* asserts if *Serial_in* is 0
SC_eq_3 asserts if *Sample_counter* = 3
SC_lt_7 asserts while *Sample_counter* < 7
BC_eq_8 asserts if *Bit_counter* = 8

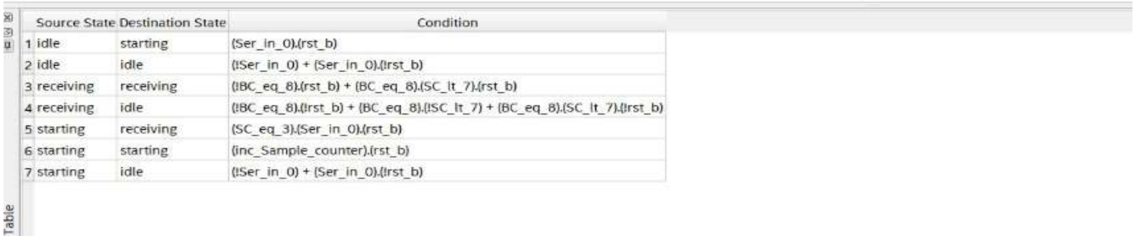
FIGURE 7-26 ASMD chart for *UART_receiver.*

FSM Diagram for Controller Rx- read directly from ASMD graph

- green text indicates this will be sent to Rx Datapath
- blue text indicates input or internal to the Rx Controller
- red text indicates output to the Transmitter (Tx)



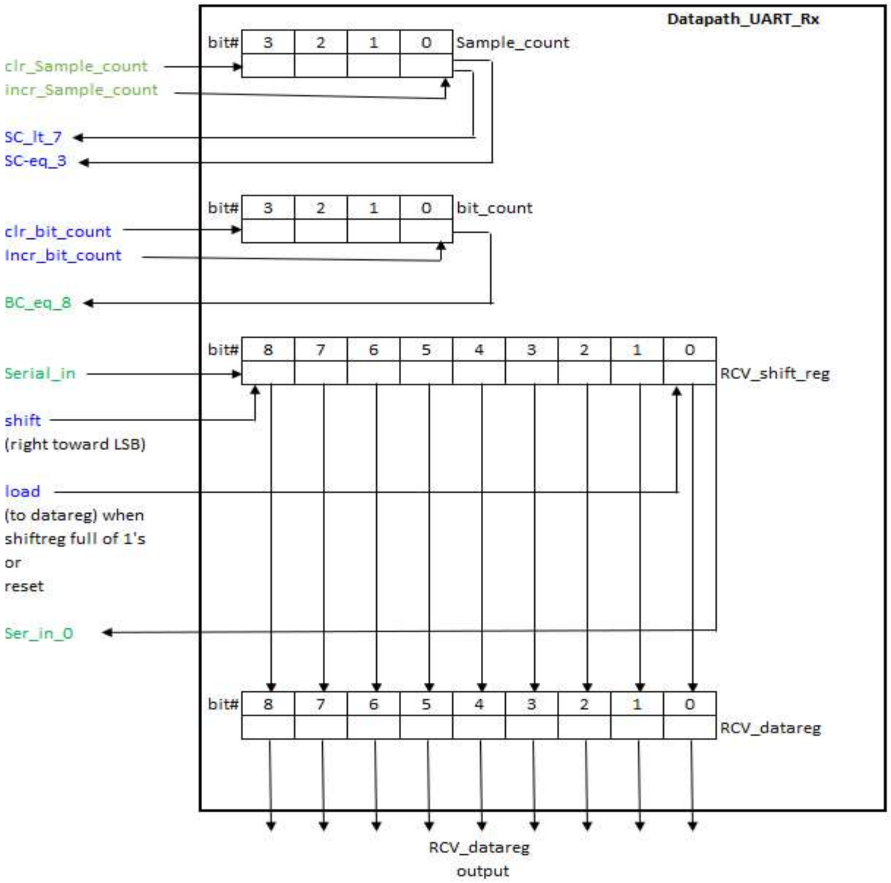




	Source State	Destination State	Condition
1	idle	starting	(Ser_in_0).(rst_b)
2	idle	idle	(!Ser_in_0) + (Ser_in_0).(!rst_b)
3	receiving	receiving	(!BC_eq_8).(rst_b) + (BC_eq_8).(SC_lt_7).(rst_b)
4	receiving	idle	(!BC_eq_8).(!rst_b) + (BC_eq_8).(!SC_lt_7) + (BC_eq_8).(SC_lt_7).(!rst_b)
5	starting	receiving	(SC_eq_3).(Ser_in_0).(rst_b)
6	starting	starting	(inc_Sample_counter).(rst_b)
7	starting	idle	(!Ser_in_0) + (Ser_in_0).(!rst_b)

Datapath Register Diagram

DataPath register data is read directly from the given ASMD graph (flowchart similar graph).


Datapath_UART_Rx
bit#
3
2
1
0
Sample_count
clr_Sample_count
incr_Sample_count
SC_lt_7
SC-eq_3
bit#
3
2
1
0
bit_count
clr_bit_count
Incr_bit_count
BC_eq_8
bit#
8
7
6
5
4
3
2
1
0
Serial_in
RCV_shift_reg
shift
(right toward LSB)
load
(to datareg) when
shiftreg full of 1's
or
reset
Ser_in_0
bit#
8
7
6
5
4
3
2
1
0
RCV_datareg
RCV_datareg
output