SPECIFICATON

Andreas Siggelkow
Circuit Design

Swarnim Man Dangol(32283)
Anand Chaudhary (32232)

Contents

1 History

DATE	UPDATES
08.11.2020	Document started
09.11.2020	Document updated with Block diagram and Description
16.01.2021	redid requirements and new block diagram
17.01.2021	uploaded block diagrams and requried FSM's
18.01.2021	Description of OSI-Layers

2 Overview

The ASIC FPGA should be able to count the number of people entering a certain place with the help of three light curtains near the door, Also there is only doorway to come and exit. There are LED's that signal if the room can be entered,**Green** LED being **GO** (The room has not reached maximum capacity) and ,**Red** LED being **STOP** (The room has reached maximum capacity). The LED's are also accompanied by acoustic sounds that indicate if the room is full or if a person has entered the room.

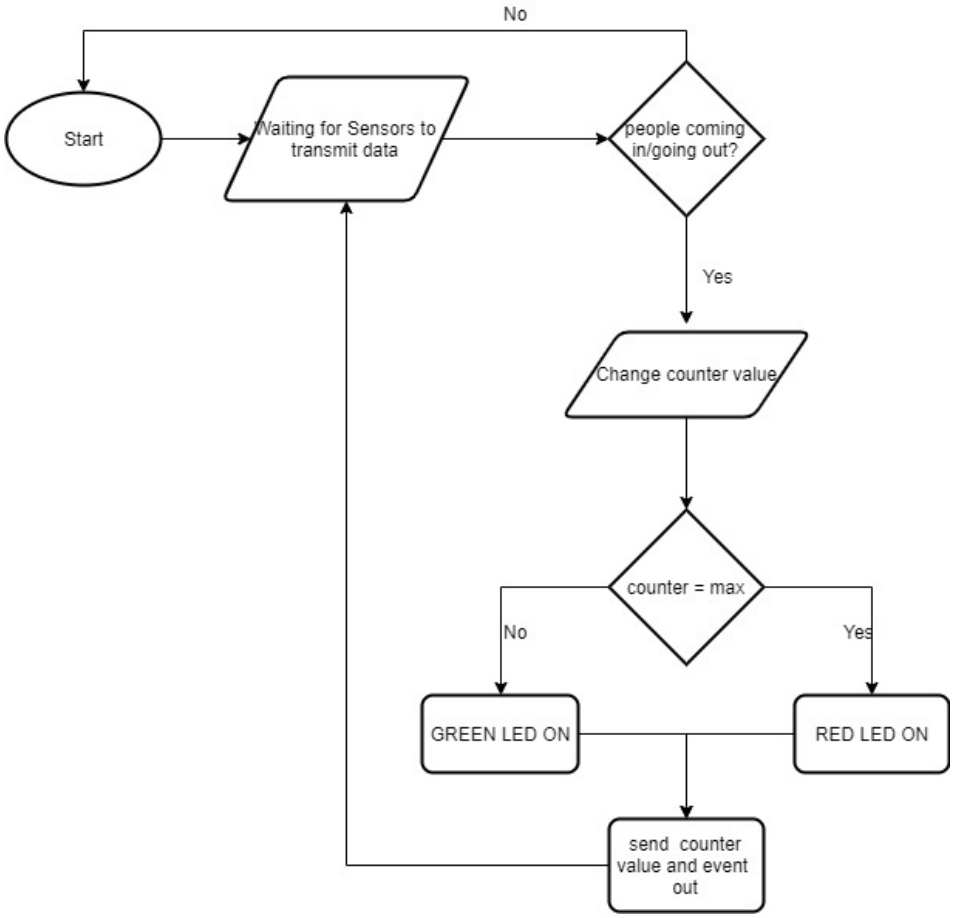


Fig 2 Flow-chart of design

3 Requirements

3.1 General

ID	Requirement	Priority	Verifiable	Description
1	#persons	High	Testbench	The number of persons in a room must be known.
2	max	High	Testbench	The number of persons in a room must not exceed a given limit.
3	only one pers.	High	N/A	Only one person can either enter or leave the room at a time.
4	three light sensors	Medium	Testbench	Along the doorway, there are three light-curtains to allow direction-tracking of possible visitors.
5	only one door	High	N/A	Only one door exists.
1	RED-LED	High	Testbench	The maximal number of persons reached.
2	GREEN-LED	High	Testbench	The maximal number of persons not reached.
1	Entered-Sound	High	N/A	A person entered the room, play a unique sound.
2	Left-Sound	High	N/A	A person left the room, play a unique sound.
3	Stop-Sound	High	N/A	The room is full, play a unique sound.

3.2 UART

ID	Requirement	Priority	Verifiable	Description
1	9600 baud	High	Testbench	The speed of the serial transmission should be set to 9600 baud.
2	8 bit	High	Testbench	The data width of the serial transmission should be set to 8 bit.
3	no parity	High	Testbench	The serial transmission should not be checked with a parity bit.
4	one stop bit	High	Testbench	The serial transmission should have only one stop bit.
6	#persons	High	Testbench	The #persons should be transmitted to a PC.

3.3 PC and S3Interface

1	PC: language	Medium	N/A	Information should be displayed on a PC, the language is C++.
2	PC: timestamp	Low	N/A	Every event should have a unique timestamp.
1	S3: interface	Low	Testbench	Use a three wire IF.
2	S3: events	Low	Testbench	All events should be transmitted via the three wire IF.
3	S3: #persons	Low	Testbench	The #persons should be transmitted via the three wire IF.

4 Architectural Concepts

4.1 Synchronous circuit

The blocks designed are all synchronized with the clock so the changes occur only when there is a rising edge of the clock.

4.2 Reset

In this design the blocks uses an active low reset signal as the built in reset buttons on the MAX10 boards are active low signals.

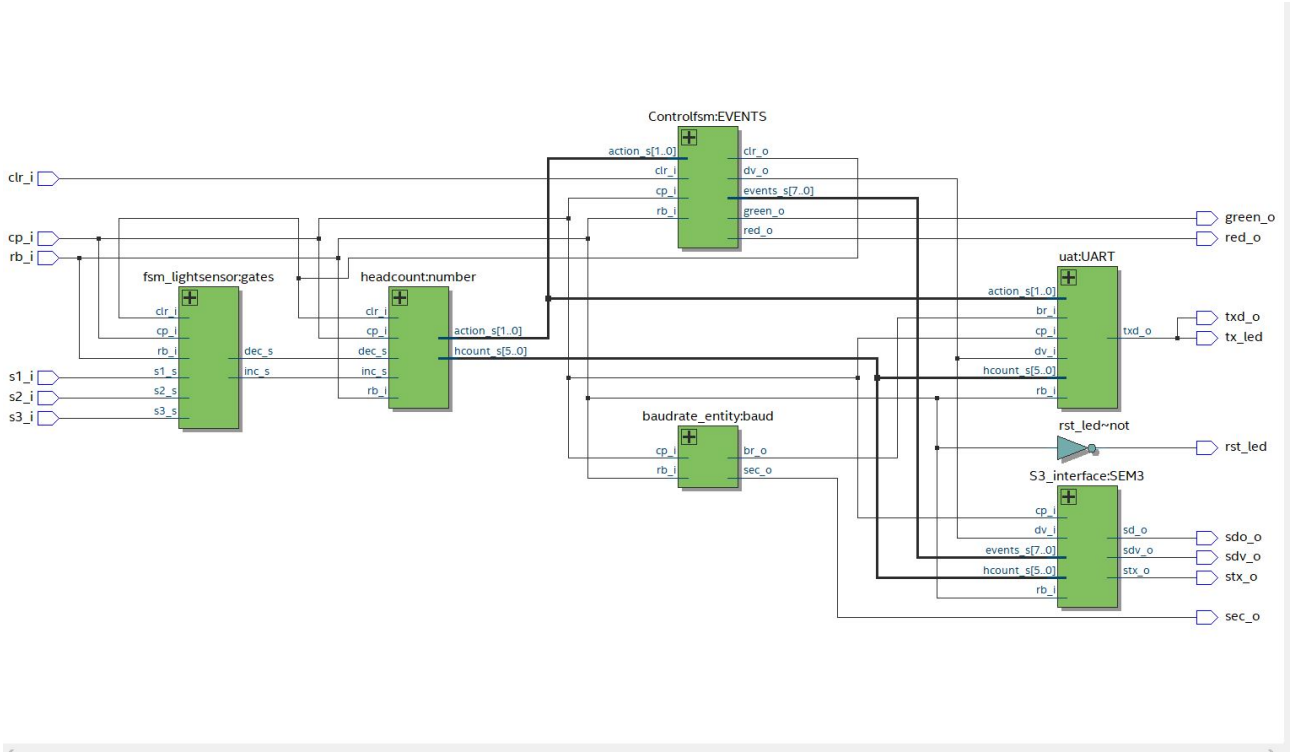
4.3 Three Process Architecture

The VHDL design, especially FSM's are built with three different processes(state transition, clock trigger, outputs) for easy readability of the code and for the simplification of the code.

4.4 Moore FSM

The VHDL design specifically uses a Moore FSM structure in every FSM present in the design, as the outputs are only affected by the change in states, It simplifies the code required to construct a FSM.

5 Top-level diagram



5.1 Input and Output Signals

Signal	Full name	Type	Description
cp_i	System Clock	I	System clock of 10 MHz
rb_i	Reset	I	reset, active low
s1_i	first sensor	I	signal from first sensor
s2_i	Second sensor	I	signal from second sensor
s3_i	thrid sensor	I	signal from third sensor
clr_i	Clear counter	I	clear and restart headcount
red_o	Red LED	O	enables red LED
grn_o	Green LED	O	enables green LED
txd_o	Serial data out	O	serial data out to the PC through RS232
sdo_o	serial data output	O	drives S3 or µC
sdv_o	serial data valid	O	drives S3 or µC
stx_o	serial data transfer active	O	drives S3 or µC
tled_o	txd-LED	O	LED on when serial data is being transferred

5.2 Internal Signals

Signal	Full name	Type	Description
br_s	Baudrate signal	S	Baudrate
inc_s	increment counter	S	high when a person walks in
dec_s	decrements counter	S	high when a person goes out
clr_i	Clear counter	S	clear and restart headcount
hcount_s	Headcount	S	number of people
event_s	Event (6-bit)	S	status signal !,+,-
action_s	action(2-bit)	S	signals people coming in, going out or max number is reached
dv_s	Data valid	S	Signals UART and S3inteface to start

6 Baud-rate Generator

This block handle the dividing of the 12 MHz System-Clock to 9600 baud (br_s)for the Serial-Transmission between the UART and the PC. It also handles a pulse signal(sec_o), which is sent every second by dividing the clock with an appropriate pre-scaler value.

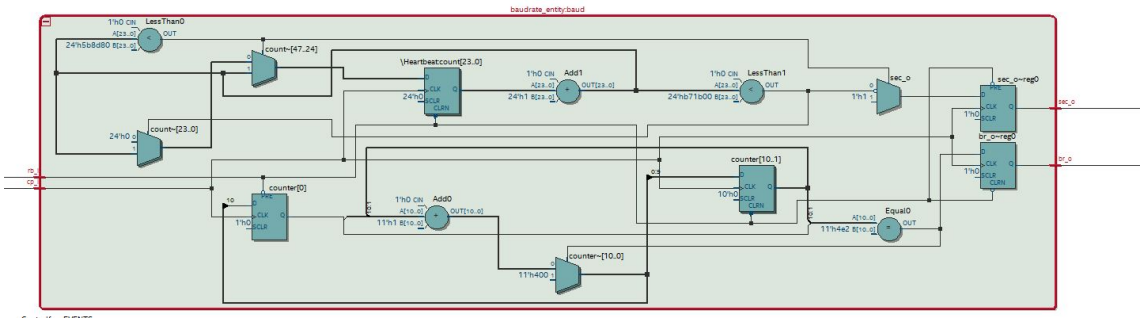


Fig 6. Shows the block of the Baud-rate Block.

Signal	Full name	Type	Description
cp_i	System Clock	I	System clock of 12 MHz
rb_i	Reset	I	reset, active low
br_s	Baudrate signal	O	Baudrate
sec_o	Heartbeat signal	O	pulse signal after every second

Table 6. (I/O) Baud-rate Block

7 Light-sensor

The light sensor block takes the input from the light-sensors and sends an output signal to the headcount block. For people coming in,it sends an increment signal (inc_s) and for people going out it sends a decrement signal(dec_s).For this, the first sensor is s1_i and the third sensor is s3_i.

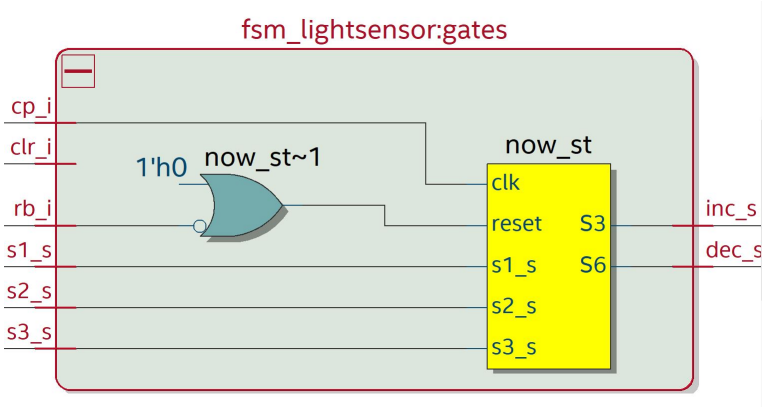


Fig 7. lightsensor block

Signal	Full name	Type	Description
cp_i	System Clock	I	System clock of 12 MHz
rb_i	Reset	I	reset, active low
s1_i	first sensor	I	signal form first sensor
s2_i	second sensor	I	signal form second sensor
s3_i	third sensor	I	signal form third sensor
inc_s	increment counter	O	high when peole walks in
dec_s	decrement counter	O	high when people goes out

Table 7.(I/O) Lightsensor block

7.1 LS FSM

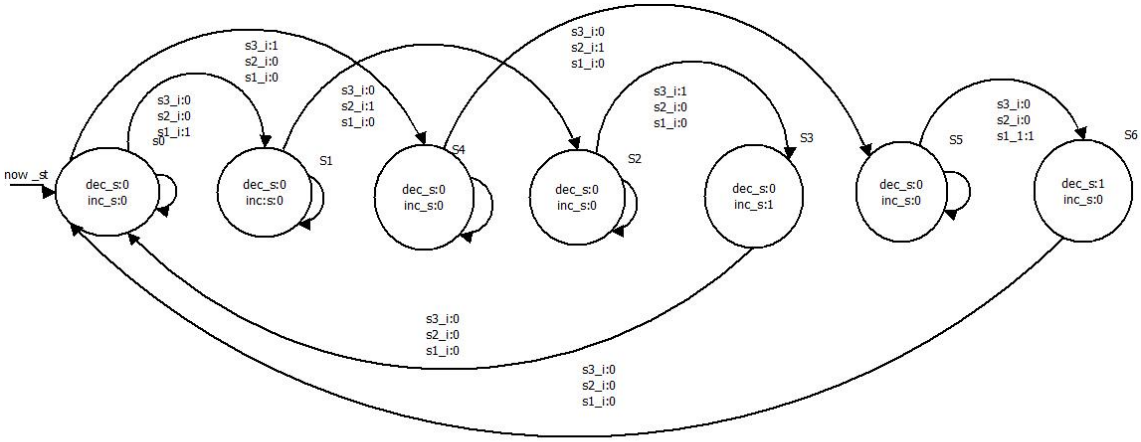


Fig: FSM of ligjt sensor Block

Fig 7.1 FSM of Ligtsensor block

State	Next State	Condition(s3_i.s2_i.s1_i)	Outputs(dec_s.inc_s)
S0(initial state)	S1	001	00
S0(initial state)	S4	100	00
S1(person enters)	S2	010	00
S2(second sensor passed)	S3	100	00
S3(Third sensor passed)	S0	000	01
S4(person leaves)	S5	010	00
S5(second sensor passed)	S6	001	00
S6(First sensor passed)	S0	000	10

7.1 State-transition table

8 Control

This block takes in incoming signals passed from the light sensor block, according to that sends the output and an enable signal to the UART and the S3interface. The outputs are selected through a Finite-State-Machine(FSM) designed in the block.

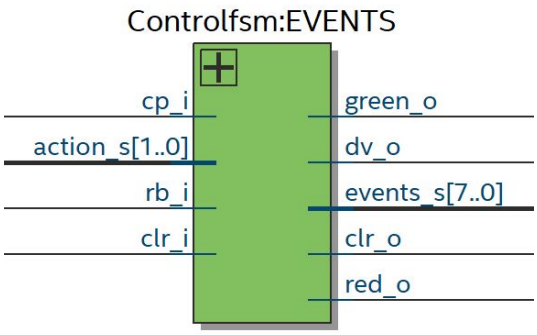


Fig 8. Controlfsm block

Signal	Full name	Type	Description
cp_i	System Clock	I	System clock of 12 MHz
rb_i	Reset	I	reset, active low
green_o	green out	O	Green led
red_o	red out	O	Red LED
action_s	action	I	signals people coming in, going out or max number is reached
event_s	event(6-bit)	O	status signal !,+,-
dv_s	Data valid	O	Signals UART and S3inteface to start
clr_i	Clear in	I	Clears everything if high
clr_o	Clear out	O	Clears everything if high

Table 8. (I/O) of the Control-Block

8.1 Control FSM

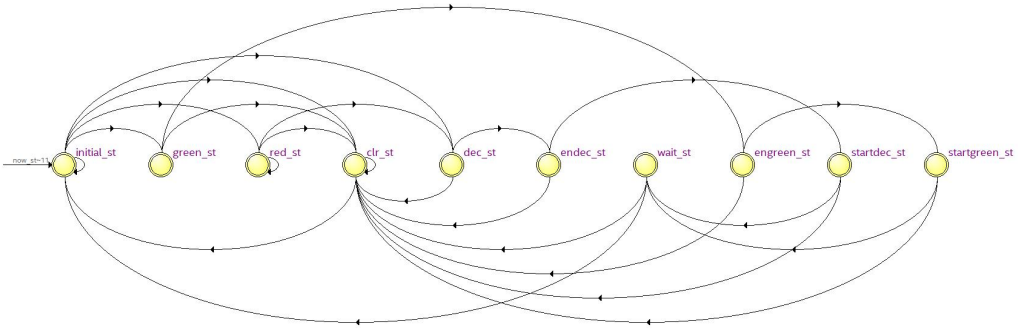


Figure 5.1 FSM Control

State	Next State	Condition (action_s)	Outputs green_s.red_s.dv_s.clr_o	events_s
initial_st	green_st	01	0000	000000
initial_st	red_st	11	000	000000
initial_st	dec_st	10	0000	000000
green_st	engreen_st	01	1000	00101011
dec_st	endec_st	10	0000	00101101
red_st	red_st	11	0110	00101011
engreen_st	startgreen_st	01	1010	00101011
endec_st	startdec_st	10	0010	00101101
startgreen_st	wait_st	01	1000	00101011
startdec_st	wait_st	10	0000	00101101
wait_st	initial_st	00	0000	00000000
clr_st	initial_st	00	0001	00000000

Table 8.1 State-transition

9 Headcount

This block with the given inputs(inc_s or dec_s) updates the number of people inside the room, by increasing and decreasing the number of people. further more it sends out an 2-bit signal to the Control block, which indicates the events taking place like people coming in, going out or the room is full or empty.

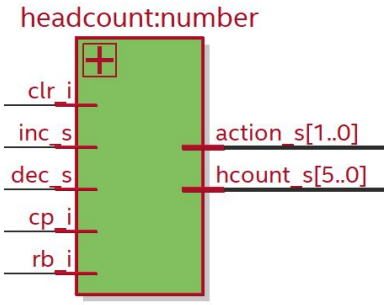


Fig.9 Headcount block

Signal	Full name	Description	
cp_i	System Clock	I	System clock of 12 MHz
rb_i	Reset	I	reset, active low
inc_s	increment counter	I	high when peole walks in
dec_s	decrement counter	I	high when people goes out
action_s	action(2-bit)	O	signals people coming in, going out or max number is reached
hcount_s	Headcount	O	number of people

Table.9 (I/O) of Headcount block

10 UART

The UART block takes in the inputs and then with the help of a fsm it loads the data to 8-bit register and transfer the given data out to serially with the help of an multiplexer. This data is sent to the PC at 9600bps, 8bits without a parity-bit and with 1 stop-bit.

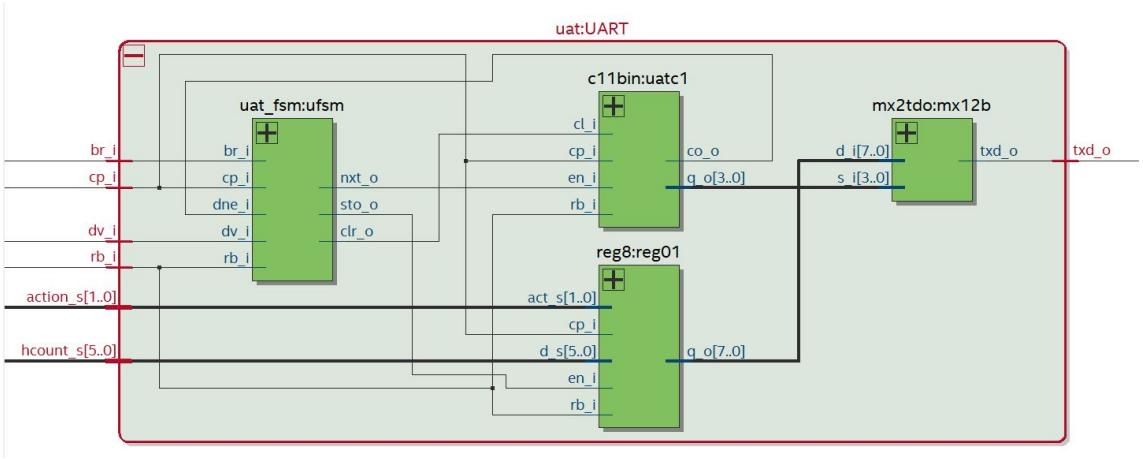
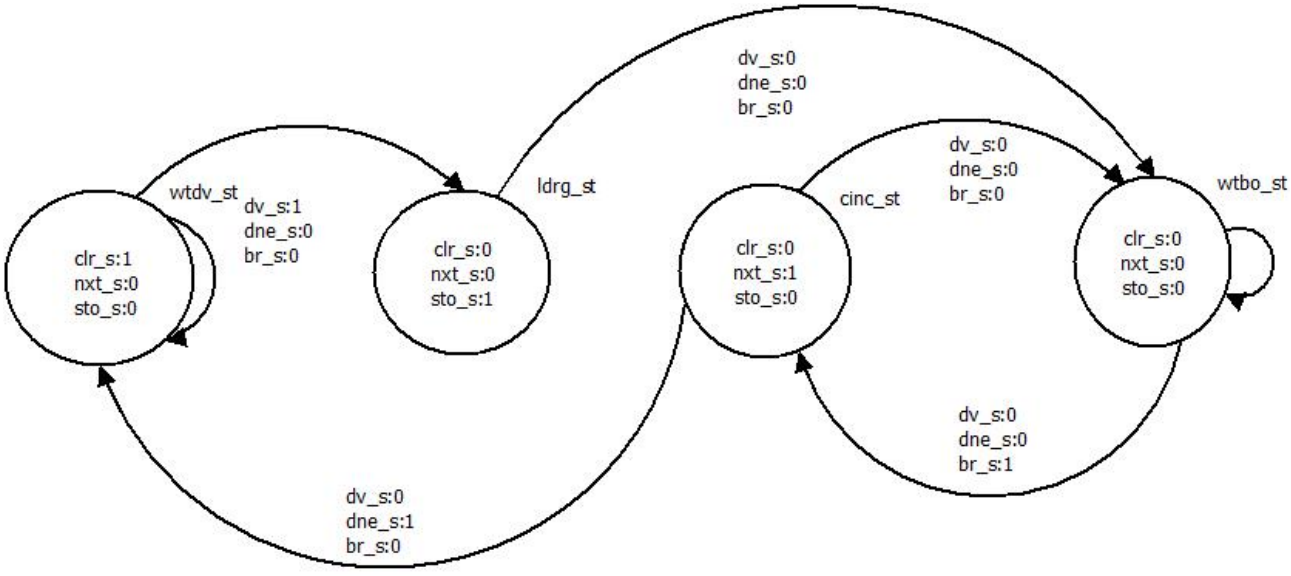


Fig 10. UART block

Signal	Full name	Type	Description
cp_i	System Clock	I	System clock of 12 MHz
rb_i	Reset	I	reset, active low
br_i	Baudrate signal	I	Baudrate
action_s	action(2-bit)	I	signals people coming in, going out or max number is reached
hcount_s	Headcount	I	number of people
clr_o	clear signal	S	clears the components inside the block
nxt_o	next bit	S	signals next bit to be transferred from the multiplexer
sto_o	load register	S	fill the register
q_o	4-bit	S	counter out
c_o	carry out	S	carry high when last bit is reached
txd_o	Serial data out	O	serial data out to the PC through RS232

Table 10. (I/O) of UART block



F FIG: UART FSM

10.1 FSM

Fig 10.1 UART FSM

State	Next State	Condition (dv_s.dne_s.br_s)	Outputs (clr_s.nxt_s.sto_s)
wtdv_st(wait for data valid)	ldrg_st	100	100
ldrg_st(load register)	wtbo_st	000	001
cinc_st(next bit)	wtdv_st	010	010
cinc_st(next bit)	wtbo_st	000	010
wtbo_st(wait for baud)	cinc_st	001	000

Table 10.1 state transition of UART FSM

10.2 Multiplexer

The Multiplexer transfer data according to the input it gains from the 4-bit counter which only counts till 12 . So for ”0000” it transfer's the first bit and soon till it reaches the end bit, after that it sends a stop signal(dne_i) to stop the fsm.

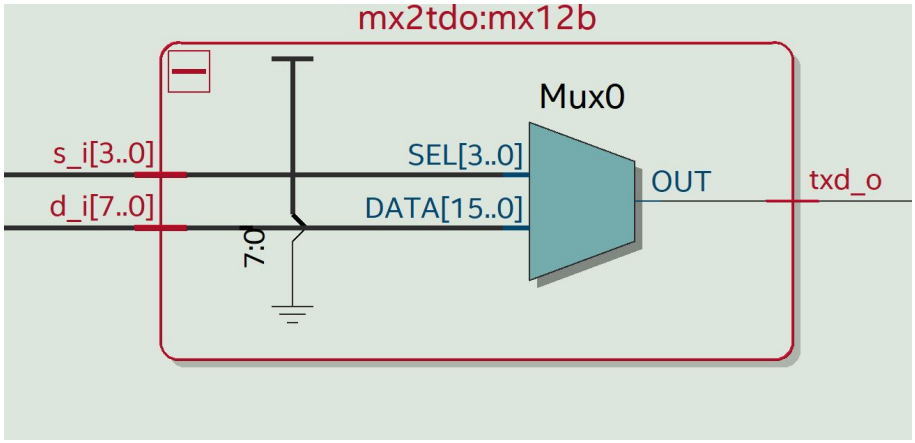


Fig 10.2 multiplexer(UART MUX)

Signal	Full name	Type	Description
s_i	4-bit selector	I	Selector
d_i	8-bit data	I	Loaded data
txd_o	1-bit out	O	Serial data out

Table 10.2 UART-multiplexer

10.3 Register

The register is an 8-bit register, storing the headcount(hcount_s) and the action(action_s). As soon as it receives the enable signal from the UART.

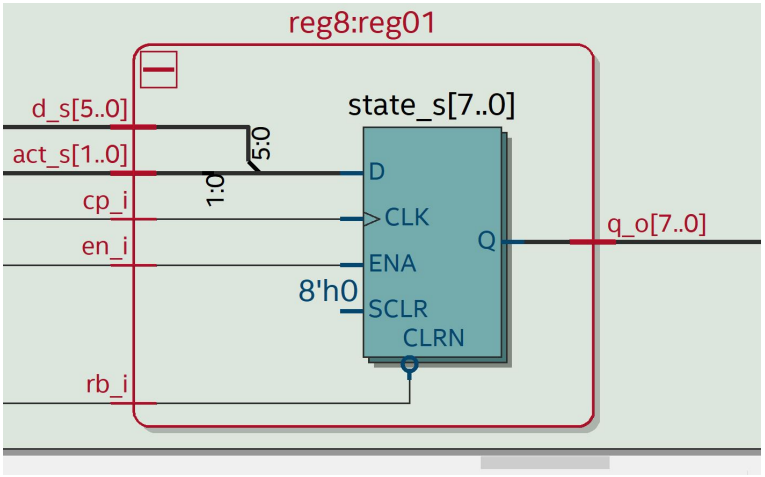


Fig 10.3 UART-register

Signal	Full name	Type	Description
cp_i	clock	I	Sys clock of 12 MHz
rb_i	reset	I	active low
en_i	enable	I	start register
d_s	6-bit data	I	number of people
action_s	2-bit data	I	+-!
q_o	8-bit data	O	stored data

Table 10.3 UART-register

11 S3 Interface

This block also like the UART takes in inputs from the Control block and with the help of a fsm loads the data to a 16-bit register and transfers the given data out to a Micro-controller. It sends out three outputs,which indicates the data trasfer is begining(stx_o), the data coming in is valid(stv_o) and the data(sto_o).

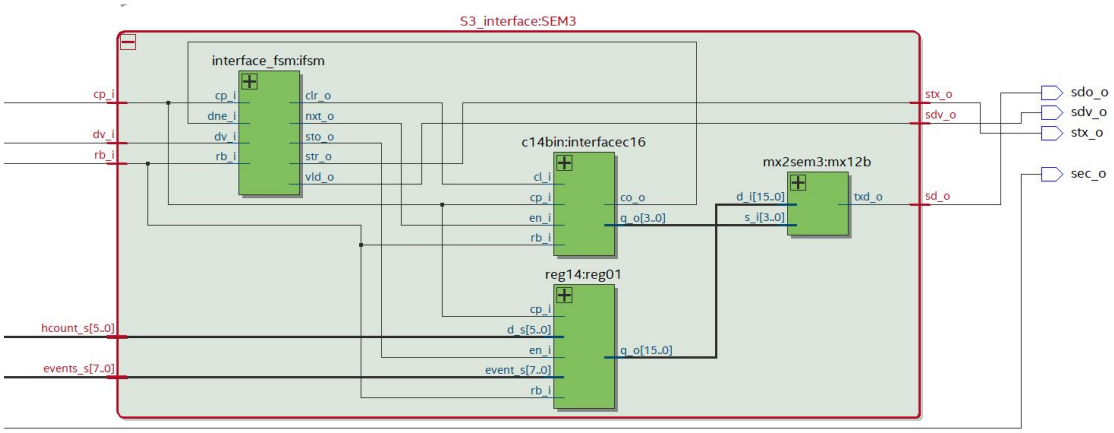


Fig 11. S3 Interface Block

Signal	Full name	Type	Description
cp_i	System Clock	I	System clock of 12 MHz
rb_i	Reset	I	reset, active low
sdo_o	serial data output	O	drives S3 or µC
sdv_o	serial data valid	O	drives S3 or µC
stx_o	serial data transfer active	O	drives S3 or µC
hcount_s	Headcount	I	number of people
event_s	Event (6-bit)	I	status signal !,+,-
dne_s	last bit reached	S	signals last bit is transferred
en_i	enable signal	S	starts the respective component
clr_o	clear signal	S	clears the components inside the block
vld_o	valid data	S	signals the out going data is valid
nxt_o	next bit	S	signals next bit to be transferred from the multiplexer
str_o	transfer start	S	out going data transfer active
sto_o	load register	S	fill the register
q_s	4-bit	S	counter out
c_o	carry out	S	carry high when last bit is reached

Table 11. (I/O) S3 Interface Block

11.1 FSM

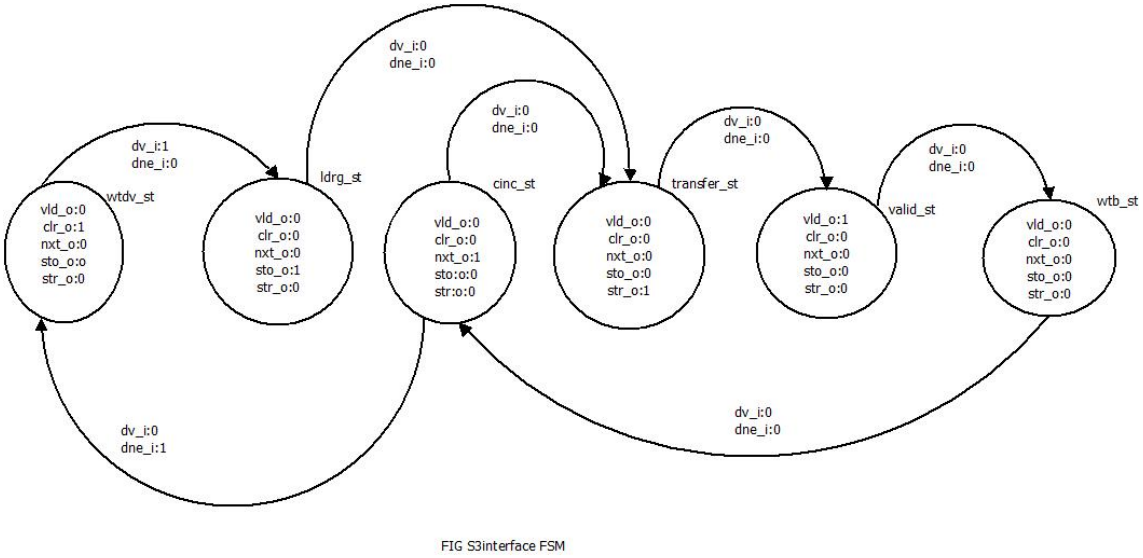


Fig 11.1 S3interface FSM

State	Next Stats	Condition (dv_i.dne_i)	Outputs (vld_o.clr_o.nxt_o.sto_o.str_o)
wtdv_st(wait for data valid)	ldrg_st	10	01000
ldrg_st(load register	transfer_st	00	00010
cinc_st(next bit	transfer_st	00	00100
cinc_st(next bit)	wtdv_st	01	00100
transfer_st(start transfer)	valid_st	00	00001
valid_st(data valid)	wtb_st	00	10000
wtb_st(wait)	cinc_st	00	00000

Table 11.1 State-transition

11.2 Multiplexer

16-bit-Mux for the S3interface. The data which needs to be sent are selected through a counter which counts to the required number of bits.

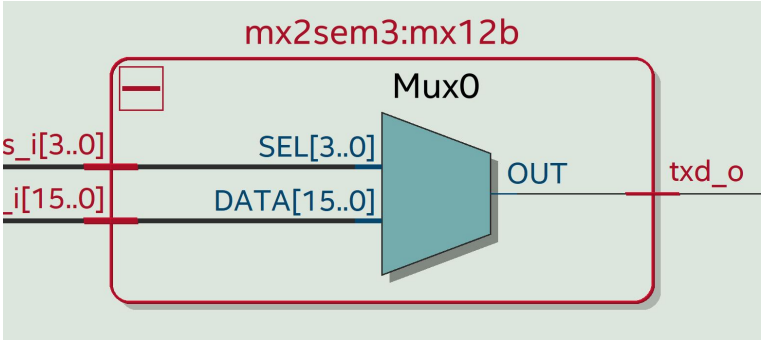


Fig 11.2 S3interface multiplexer

Signal	Full name	Type	Description
s_i	4-bit selector	I	Selector signal
d_i	16-bitdata	I	loaded data
txd_o	serial data output	O	drives S3 or µC

Table 11.2 MuxInterface

11.3 Register

The register is hold's the data passed in and waits till an enable signal(en_i) is recieved. Then it passes the data out to the multiplexer.A 16-bit register is used for the S3 interface used to store the number of people(d_s) and the event signal(event_s) being passed.

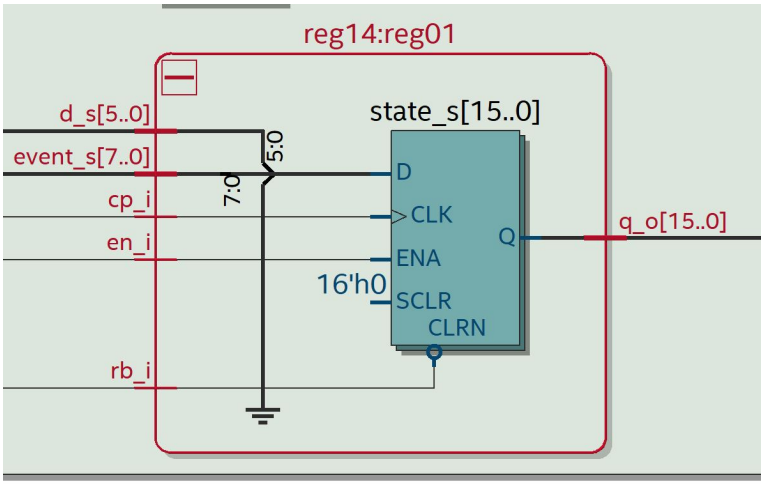


Fig 11.3 S3interface register

Signal	Full name	Type	Description
cp_i	clock	I	Sys clock of 12 MHz
rb_i	reset	I	active low
en_i	enable	I	start register
d_s	6-bit data	I	number of people
event_s	8-bit data	I	+-!
q_o	16-bit data	O	stored data

Table 11.3 S3register

12 Counter

The Baud-rate generator,UART and S3Interface consists of a counter which counts to a specific set value and when the counter reaches the specific set value sends out a pulsed signal out. The UART and The S3 interface also uses a counter but there the counter acts as a selector to select the outgoing bits from the multiplexer. For the UART and S3Interface, the counter's state changes according to an enable signal coming from the IS3 and UART FSM.

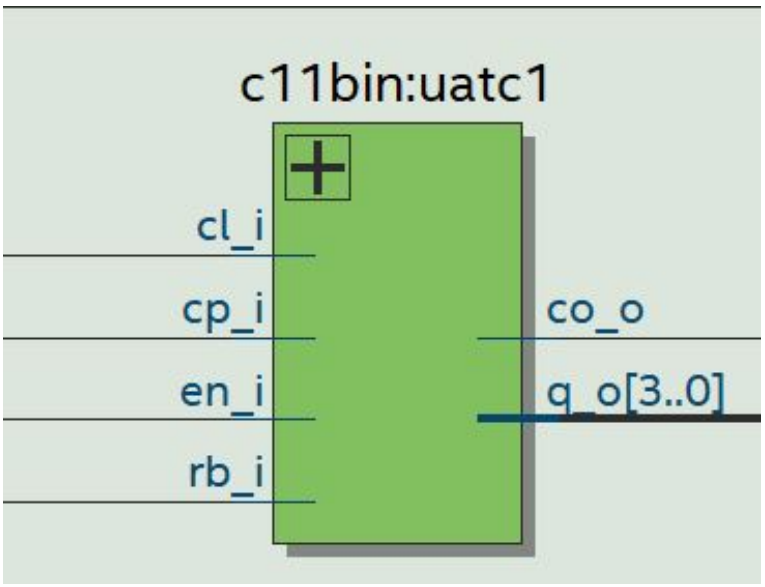


Fig 12. Counter-Component

Signal	Full name	Type	Description
cp_i	clock	I	Sys clock of 12 MHz
rb_i	reset	I	active low
en_i	enable	I	start counter
c_o	carry	O	final bit reached
q_o	3-bit	O	Selector

Table (I/O) Counter-Component

13 OSI layers

The OSI model **(OPEN SYSTEM INTERCONNECTION MODEL)** represents how the communication functionality of a system is working. There are a total of 7 layers that define the OSI model these are :

1)Physical Layer
Layer where data bits are transmitted over a physical medium

2)Data link Layer
Layer defines the data format

3)Network Layer
Layer which defines what path the data takes

4)Transport Layer
Layer where the data transmission takes place with help of protocols.

5)Session Layer
Layer which maintains the connection between ports.

6)Presentation Layer
Layer which makes sure that the data is in a presentable manner, data encryption might also occurs here.

7)Application Layer
Layer where people can access the Network-Interface through an application

These layers can be further divided into four blocks based on their description, the Physical-Layer and the Data-Link Layer can be together as Network-Interface-Block, Network-Layer as Internet-Block, Transport-Layer as Transport-Block and the last three layers (Session, Presentation, Application) as Application-Block.

Block	Layer	Implementation
Network-Interface	Physical	The UART-Block and the S3-Interface-Block readies raw data bits to be transmitted through the physical communication-medium.
	Data-Link	The data through the UART with 1- stop- bit, packages data into groups of 8-bits, since there are no parity and check-bits there is no way to know if the data is correct.
Internet	Network	Since the data is transmitted through an USB cable routing is not required.
Transport	Transport	redirecting of data is not needed
Application	Session	Serial data communication session through the computer port and the RS232 adapter.
	Presentation	Serial 8-bit data are converted to it's corresponding decimal value and presented in the C++ console.
	Application	C++ console allows the user to print out the data(number of people).