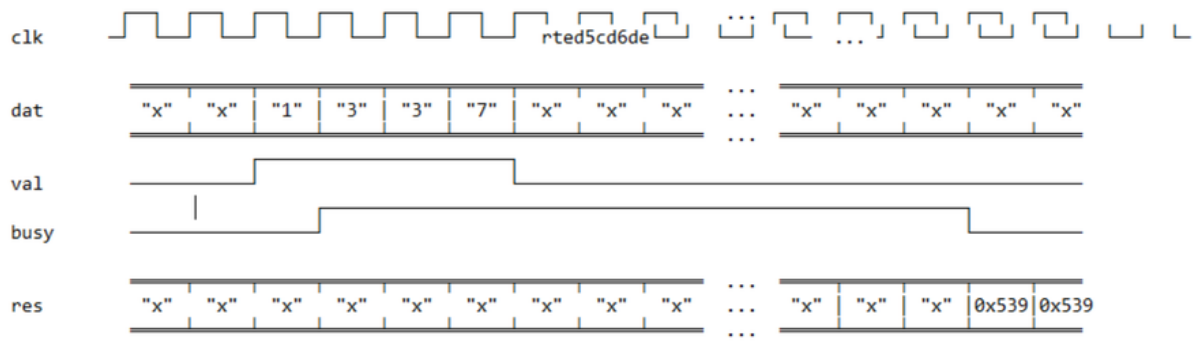
ASCII to Binary Conversion Core

Problem statement:

Consider an 8-bit bus of ASCII symbols from 0 to 9 sent as packets (`dat` and `val`) forming ASCII-encoded decimal numbers. The task is to create a module that decodes these packets into binary numbers. Please provide simulation and synthesis results for the module.

- Number of cycles it takes for the module to convert is not specified
- Consider not using multiplier
- Minimum length of the number is 1, maximum is set by a parameter (generic). E.g. 4
- for 1337. If length exceeds maximum, packet is ignored. The module should support 8 as maximum length of input decimal number (maximum value would be 99999999)
- Module should be synthesizable by Xilinx Vivado 2018 and newer and is able to pass STA at 100MHz (Spartan 7 device) with all ports mapped to I/O
- There is no need to provide hardware, only a bitstream is sufficient
- The module should not latch regardless of input and ignore all other packets while busy.
- A `busy` signal is used to indicate that conversion is in progress. It is asserted 1 tick after `val` and is de-asserted when the conversion is done. 1 tick after de-assertion, a
- result appears on the `res` (result) port.
- Sizing of the ports should depend on a parameter
- Simulation tool not specified
- 10. Language: System Verilog or VHDL
- 11. Make your assumptions wherever required



Solution:

- first covert ASCII to BCD

The input data is coming as ASCII encoded decimal numbers from 0 to 9.

The ASCII encoded table is given below

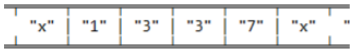
Character	ASCII Code (Decimal)	ASCII Code (Hex)	Binary Representation (8 bits)

'0'	48	0x30	00110000
'1'	49	0x31	00110001
'2'	50	0x32	00110010
'3'	51	0x33	00110011
'4'	52	0x34	00110100
'5'	53	0x35	00110101
'6'	54	0x36	00110110
'7'	55	0x37	00110111
'8'	56	0x38	00111000
'9'	57	0x39	00111001

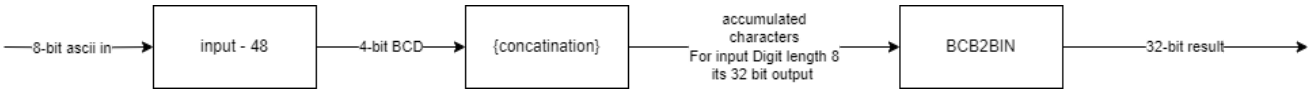
haracter ASCII Code (Decimal) Binary Representation (8 bits)

Each character is represented by an 8-bit ASCII code input to the core.

As



As the above input “1“, is considered as a character encoded as an ASCII encoded decimal which is in binary 0x31. So the approach is to convert the input data to BCD first for the next calculations.

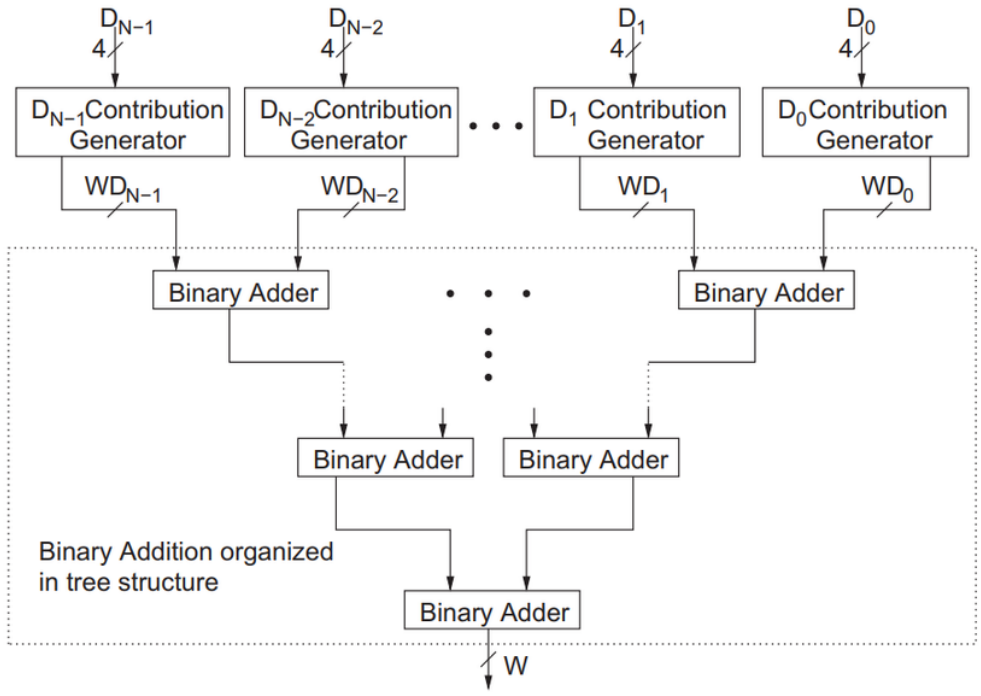


- covert BCD to binary

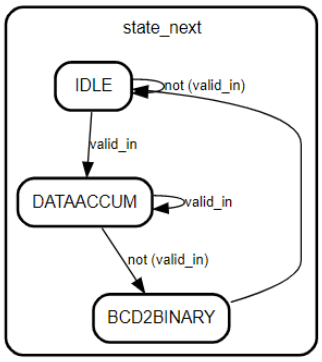
The BCD conversion is done based on 4-bit grouping defined and proposed in the below architecture

It requires the Accumulated precalculated 10’s table for each digit and used the simple adder to give the BCD to binary result. Since the timing is closed easily for 8 digits maximum number of 100Mhz and the tool has added the optimal adders by default, So no optimization is performed for Adders.

High performance FPGA-based decimal-to-binary conversion schemes for decimal arithmetic - ScienceDirect



FSM: The Top level architecture is a sequential data input and there is NO stringent requirement for minimum clock cycles to be used, a simple FSM-based implementation is used for better control on signals and flow.



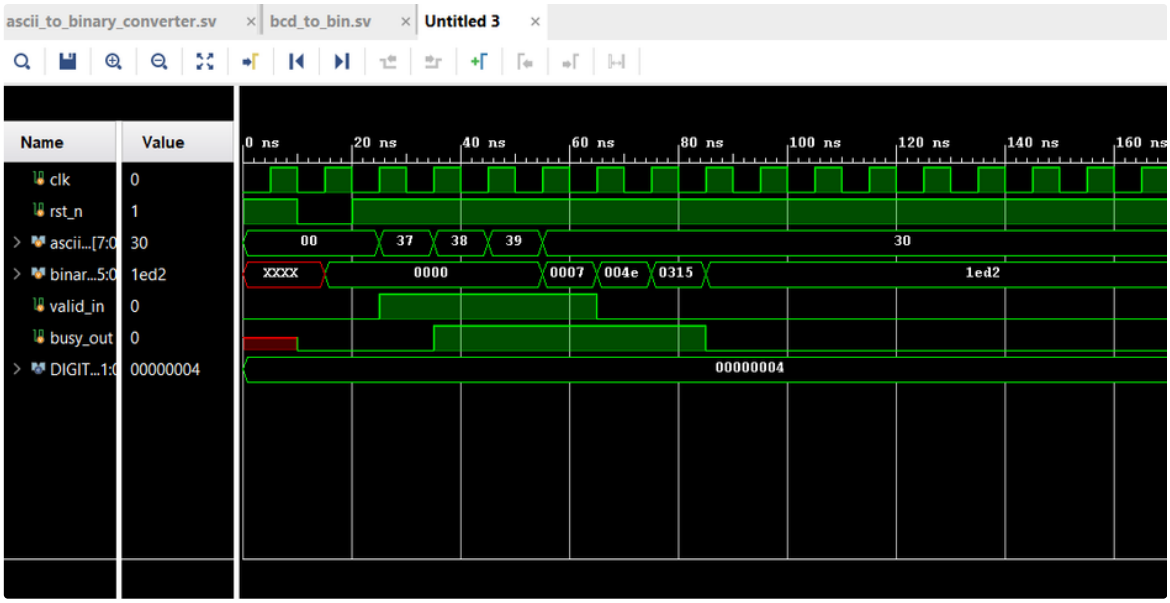
SIMs:

DIGITS = 4

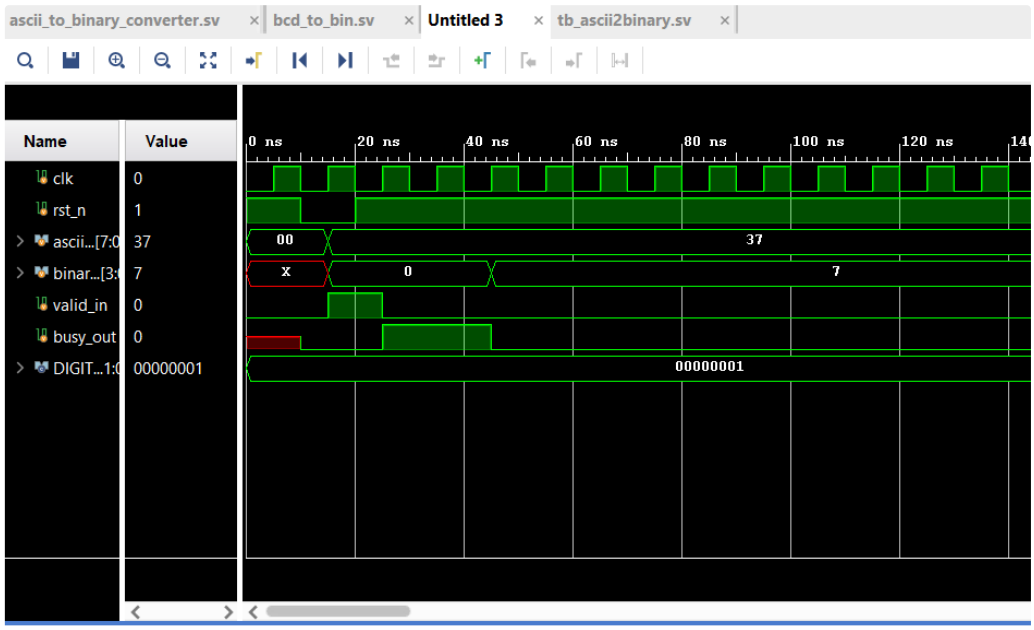
the input data is shown as Hex values in the below waveforms.

The input number below is “7“,”8”,”9”,”0” as ascii encoded inputs.

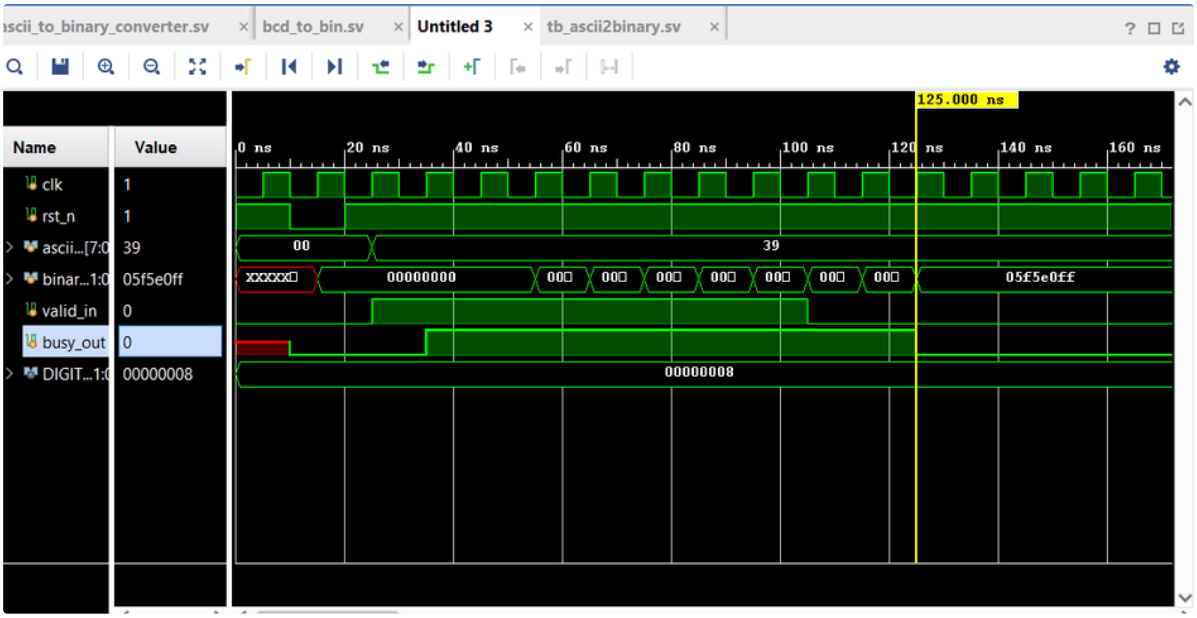
The result for 7890 decimal number converted to binary is 0x1ED2, and its present after busy signal.



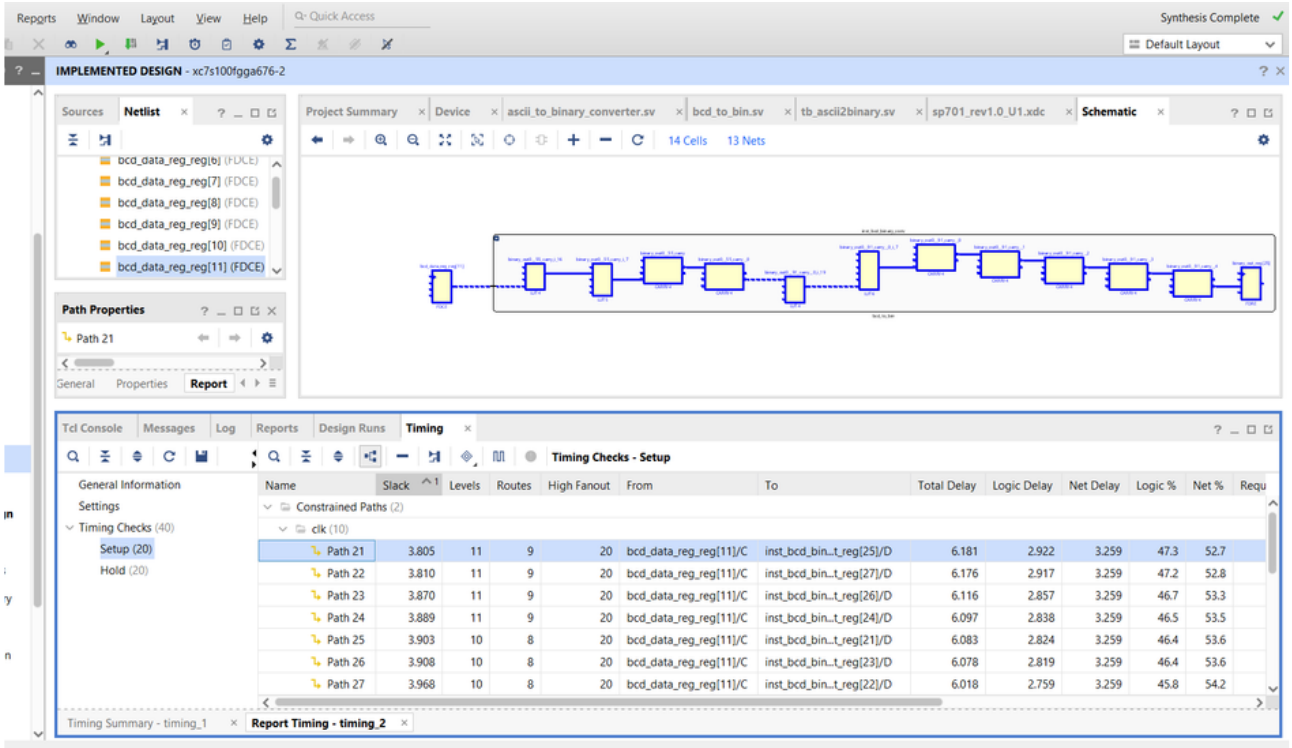
DIGITS = 1;



DIGITS = 8;


ascii_to_binary_converter.sv
bcd_to_bin.sv
Untitled 3
tb_ascii2binary.sv
Name
Value
clk
1
rst_n
1
ascii...[7:0]
39
binar...1:0]
05f5e0ff
valid_in
0
busy_out
0
DIGIT...1:0]
00000008
125.000 ns
0 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns
140 ns
160 ns
00
39
XXXXXX
00000000
05f5e0ff
00000008

STA:


Reports
Window
Layout
View
Help
Quick Access
Synthesis Complete
Default Layout
IMPLEMENTED DESIGN - xc7s100fgga676-2
Sources
Netlist
bcd_data_reg_reg[7] (FDCE)
bcd_data_reg_reg[8] (FDCE)
bcd_data_reg_reg[9] (FDCE)
bcd_data_reg_reg[10] (FDCE)
bcd_data_reg_reg[11] (FDCE)
Path Properties
Path 21
General
Properties
Report
Project Summary
Device
ascii_to_binary_converter.sv
bcd_to_bin.sv
tb_ascii2binary.sv
sp701_rev1.0_U1.xdc
Schematic
14 Cells
13 Nets
Tcl Console
Messages
Log
Reports
Design Runs
Timing
Timing Checks - Setup
General Information
Settings
Timing Checks (40)
Setup (20)
Hold (20)
Name
Slack
Levels
Routes
High Fanout
From
To
Total Delay
Logic Delay
Net Delay
Logic %
Net %
Constrained Paths (2)
clk (10)
Path 21 3.805 11 9 20 bcd_data_reg_reg[11]/C inst_bcd_bin...t_reg[25]/D 6.181 2.922 3.259 47.3 52.7
Path 22 3.810 11 9 20 bcd_data_reg_reg[11]/C inst_bcd_bin...t_reg[27]/D 6.176 2.917 3.259 47.2 52.8
Path 23 3.870 11 9 20 bcd_data_reg_reg[11]/C inst_bcd_bin...t_reg[26]/D 6.116 2.857 3.259 46.7 53.3
Path 24 3.889 11 9 20 bcd_data_reg_reg[11]/C inst_bcd_bin...t_reg[24]/D 6.097 2.838 3.259 46.5 53.5
Path 25 3.903 10 8 20 bcd_data_reg_reg[11]/C inst_bcd_bin...t_reg[21]/D 6.083 2.824 3.259 46.4 53.6
Path 26 3.908 10 8 20 bcd_data_reg_reg[11]/C inst_bcd_bin...t_reg[23]/D 6.078 2.819 3.259 46.4 53.6
Path 27 3.968 10 8 20 bcd_data_reg_reg[11]/C inst_bcd_bin...t_reg[22]/D 6.018 2.759 3.259 45.8 54.2
Timing Summary - timing_1
Report Timing - timing_2