Lab 3 - Where No Clock Has Gone Before

Embedded Systems 1
Lab Report 3
Tim Petersen - trp87
3/23/2020

Part 1:

Purpose

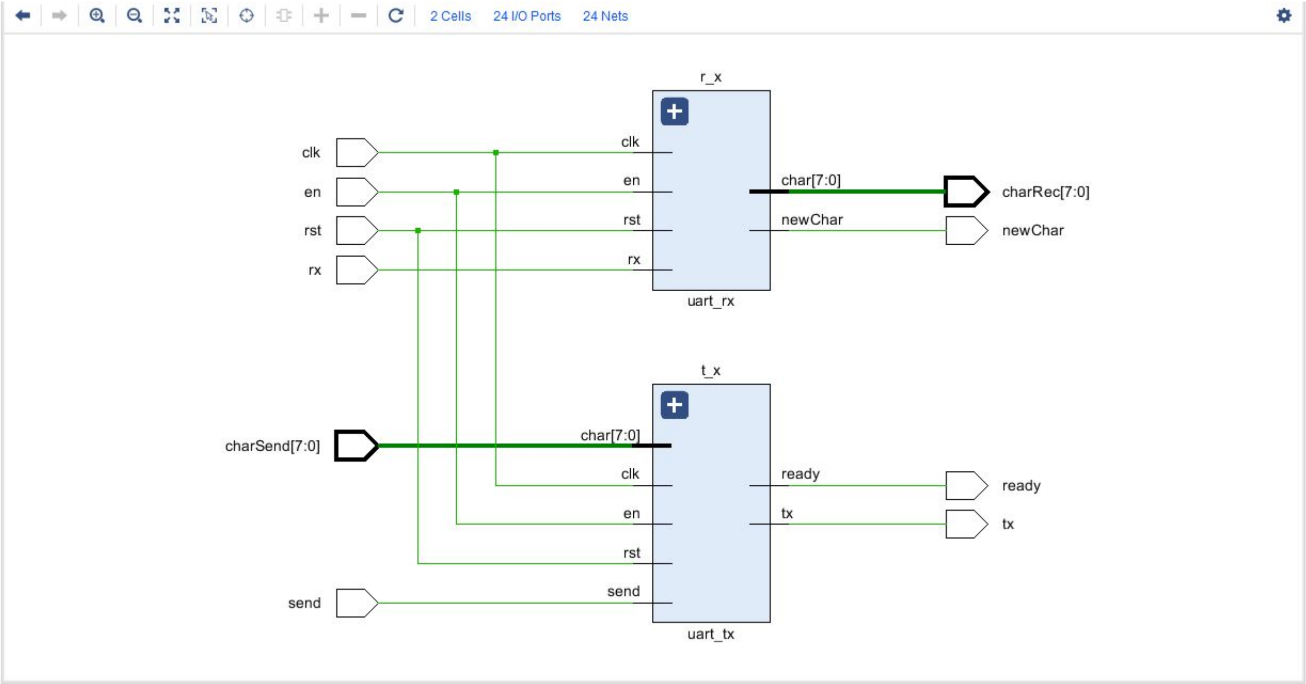
Create an entity called uart tx with the following interface that sends data using the protocol in figure (3.1) with 8 data bits, 0 parity bits, and 1 stop bit. See listing (3.1) for entity description. It should have the following behavior:

- When rst is asserted, all internal registers are cleared and it goes into an idle state
- When it is in the idle state, ready is 1
- When send is asserted and enable is 1 and the clock is on a rising edge, it stores char into a register and begins the sequence of sending data

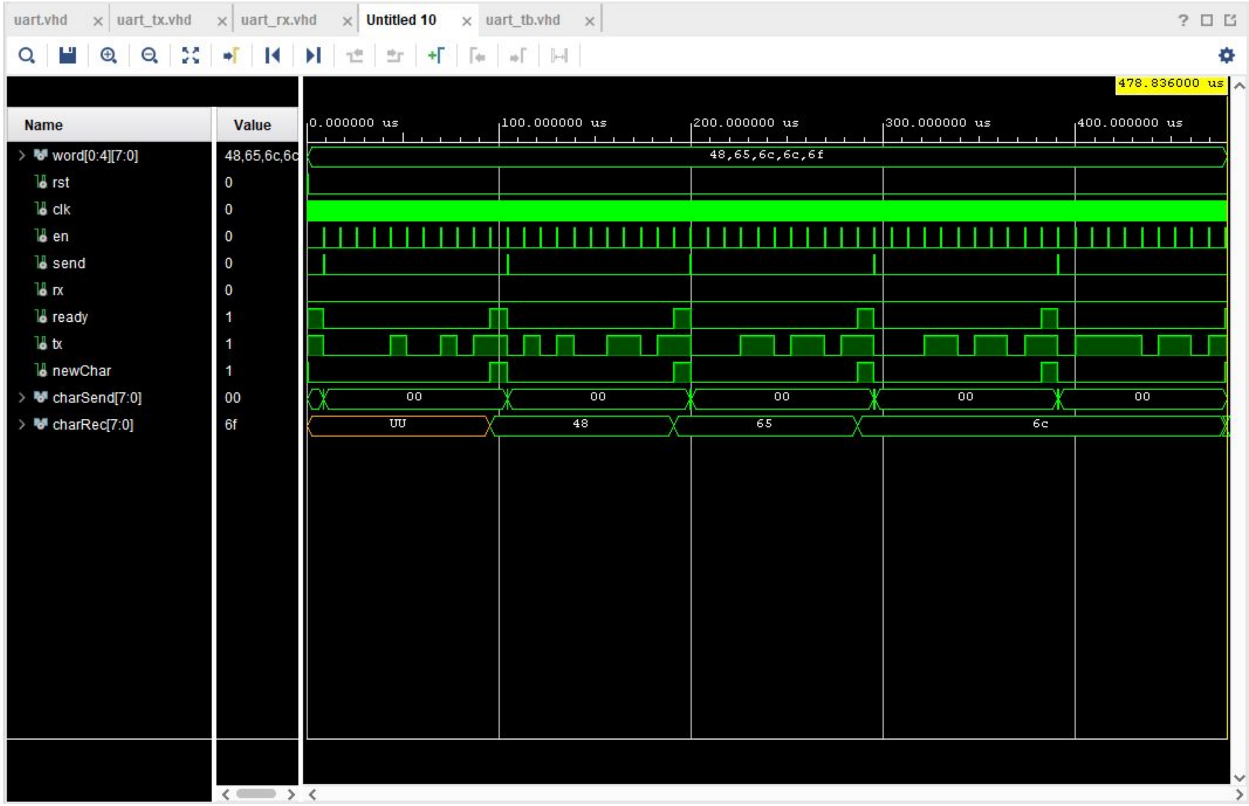
Uart_tx.vhd

Code is on github: trp87-rutgers

RTL Schematic

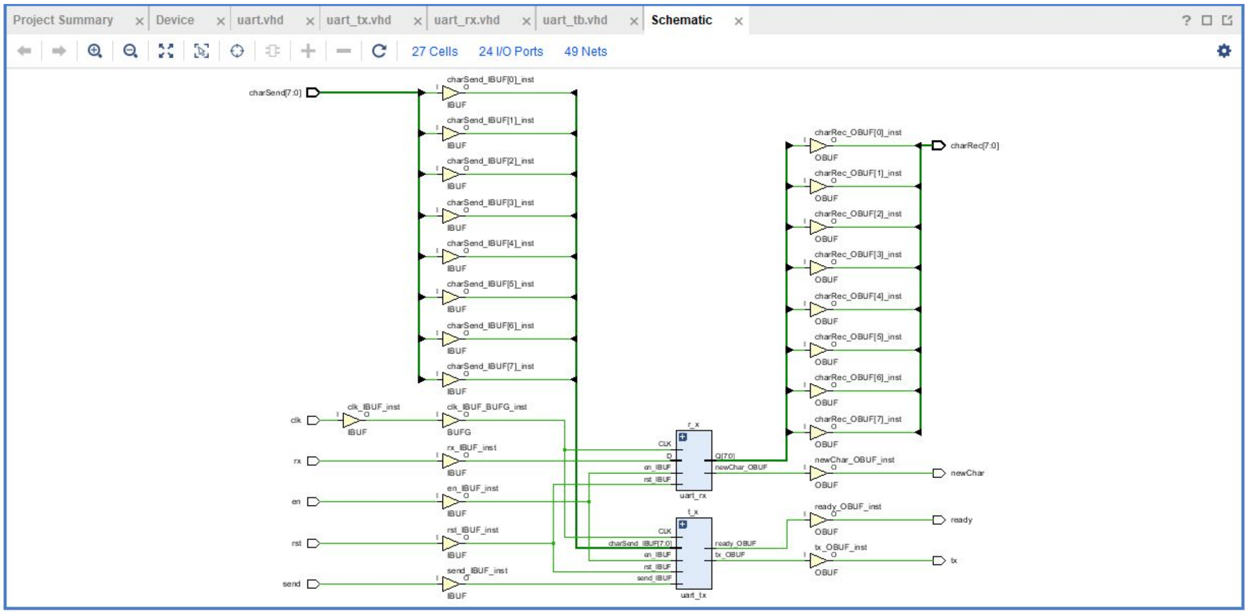


Simulation Waveform



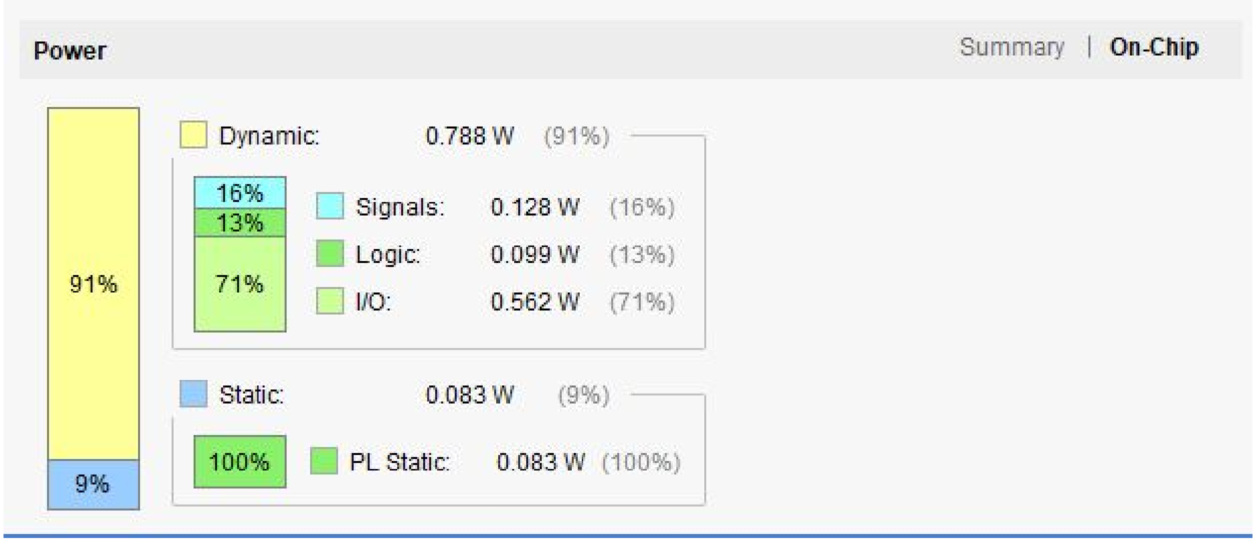
Name	Value
word[0:4][7:0]	48,65,6c,6c
rst	0
clk	0
en	0
send	0
rx	0
ready	1
tx	1
newChar	1
charSend[7:0]	00
charRec[7:0]	6f

Synthesis Schematic:



Post Synthesis Utilization Table

On Chip Power Graph



Power

🟨 Dynamic:	0.788 W	(91%)
🟦 Signals:	0.128 W	(16%)
🟩 Logic:	0.099 W	(13%)
🟩 I/O:	0.562 W	(71%)
🟦 Static:	0.083 W	(9%)
🟩 PL Static:	0.083 W	(100%)

Part 2:

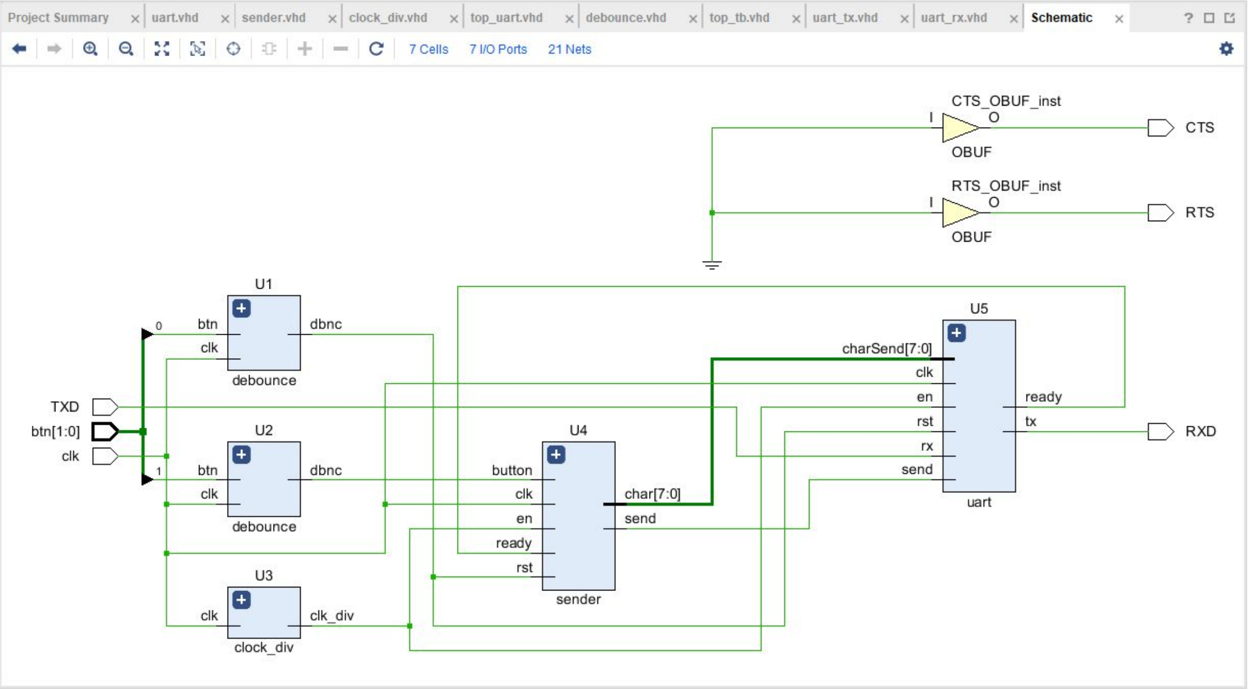
Purpose

Purpose of this part is to successfully combine the previous debounce, and modified clock div, with the new sender, and uart to communicate ‘trp87’ serially over a wire RXD.

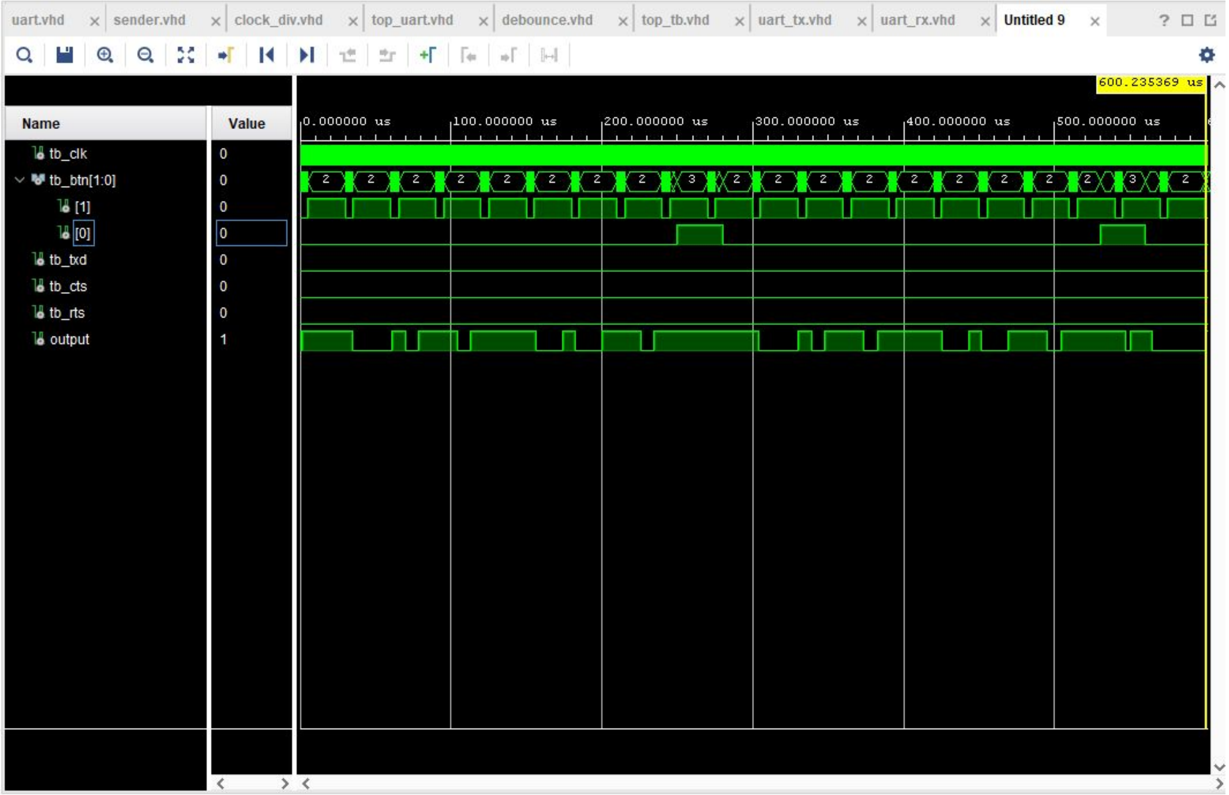
Top_uart.vhd

Code is on github for all files including test bench: trp87-rutgers

RTL Schematic

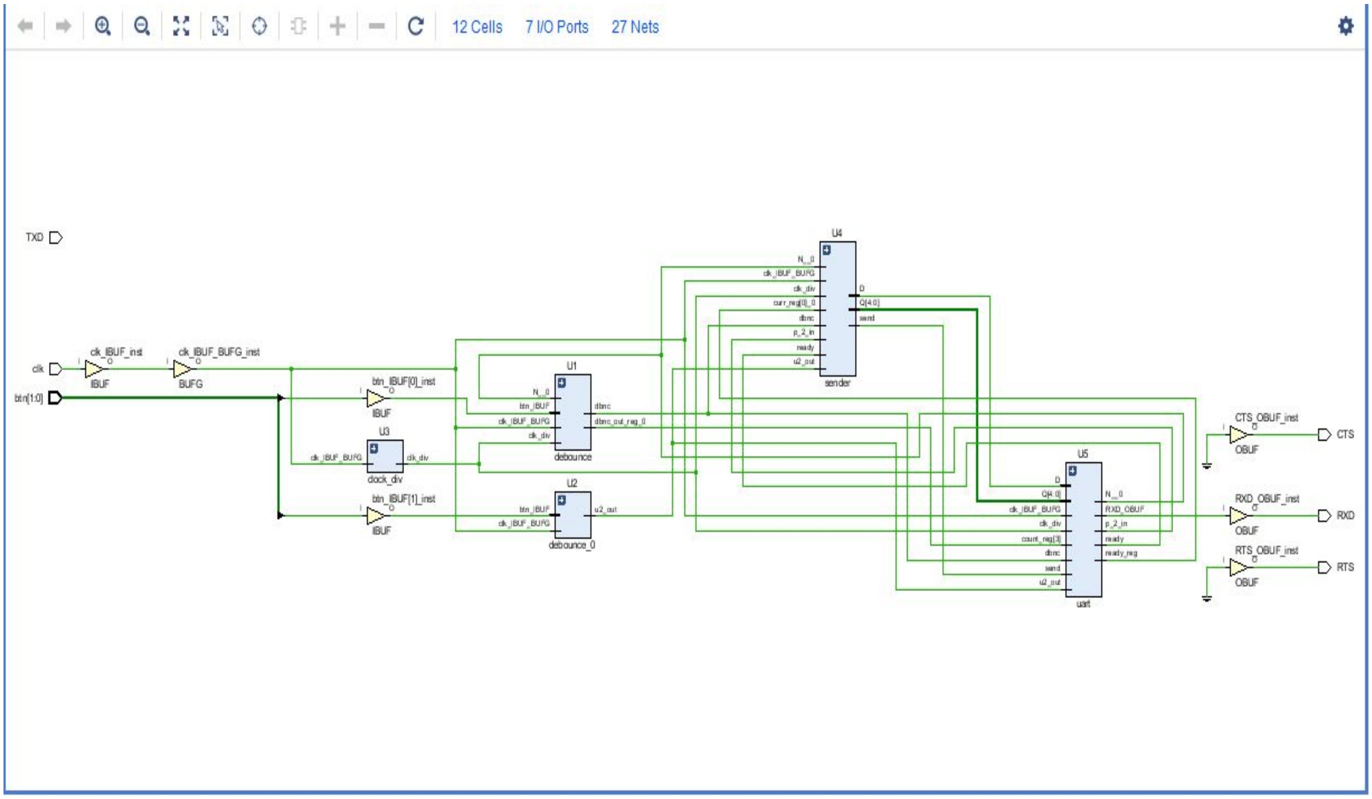


Simulation Waveform

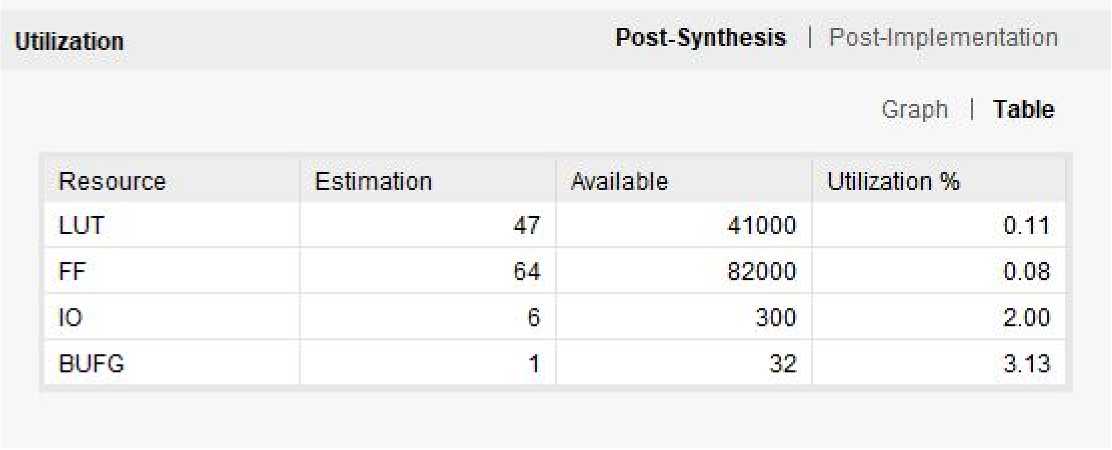


Name	Value
tb_clk	0
tb_btn[1:0]	0
[1]	0
[0]	0
tb_txd	0
tb_cts	0
tb_rts	0
output	1

Synthesis Schematic:



Post Synthesis Utilization Table



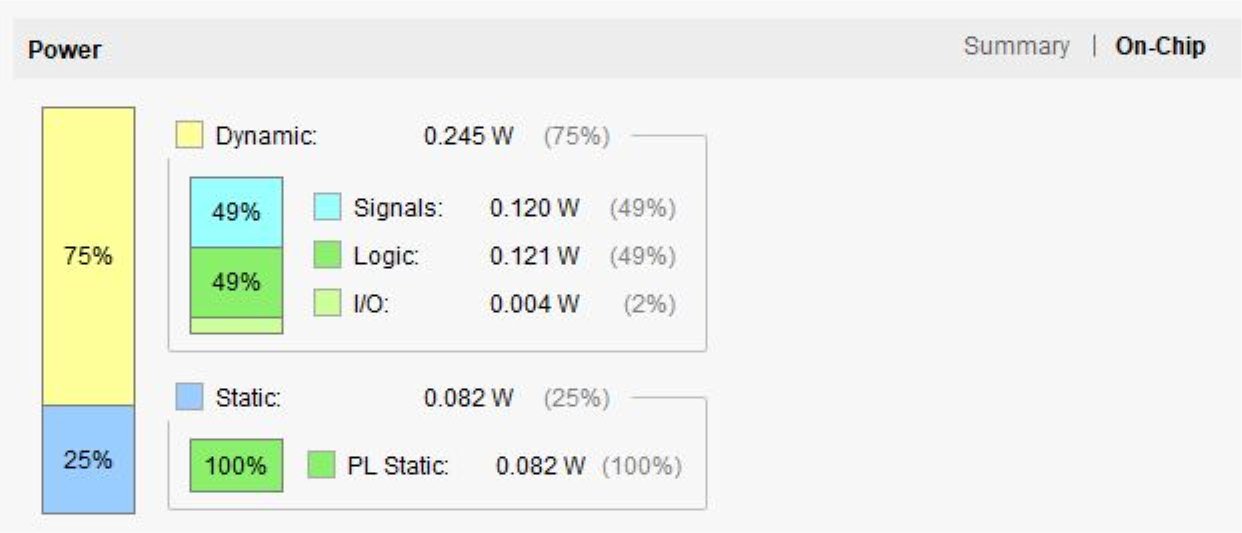
Utilization

Post-Synthesis | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	47	41000	0.11
FF	64	82000	0.08
IO	6	300	2.00
BUFG	1	32	3.13

On Chip Power Graph



```
🟨 Dynamic:      0.245 W   (75%)
    🟦 Signals:    0.120 W   (49%)
    🟩 Logic:      0.121 W   (49%)
    🟩 I/O:        0.004 W    (2%)
🟦 Static:       0.082 W   (25%)
    🟩 PL Static:  0.082 W  (100%)
```

I learned how to successfully implement a UART communication device, and using my first PMOD in a project (‘sort of’).