OpenRISC hardware bootloader over WiFi

Germán Cano Quiveu

Universidad de Sevilla
Departamento Tecnología Electrónica
(University of Seville - DTE)

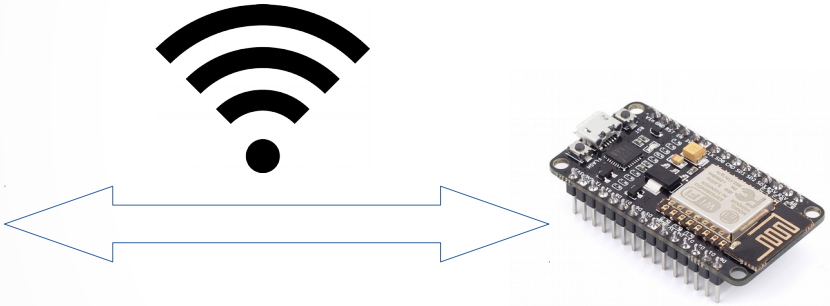
Background

- TIC204 : Research group, lines of work
  - Design and verification of digital integrated circuits.
  - Design of digital embedded systems
  - Design of SoC (System on Chip) on programmable devices (FPGA).
  - Design of application-specific peripherals for SoC.
  - Design of digital systems based on open hardware and software.

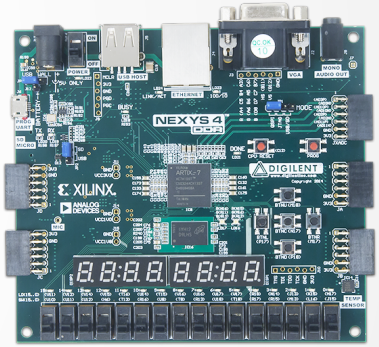
General Idea

SERVER

NODEMCU



SoC-OpenRISC



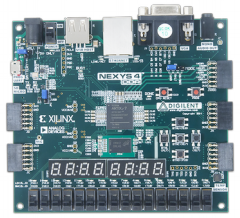
CREATE A IoT NODE THAT CAN BE UPDATED BY WIFI

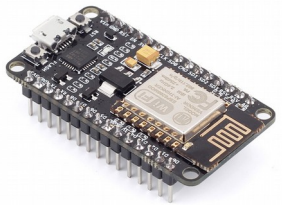
Main Goals

- Update any code anywhere.
- Use the minimum amount of resources.
- Try to utilize a modular design.

Main Actors

- SoC-OpenRISC
  - Nexys4DDR
  - Verilog
- NodeMCU
  - ESP8266: WiFi, TCP/IP
  - Arduino IDE
- Server
  - Send Binary Code
  - Python Scripts





NEXYS4DDR PORT

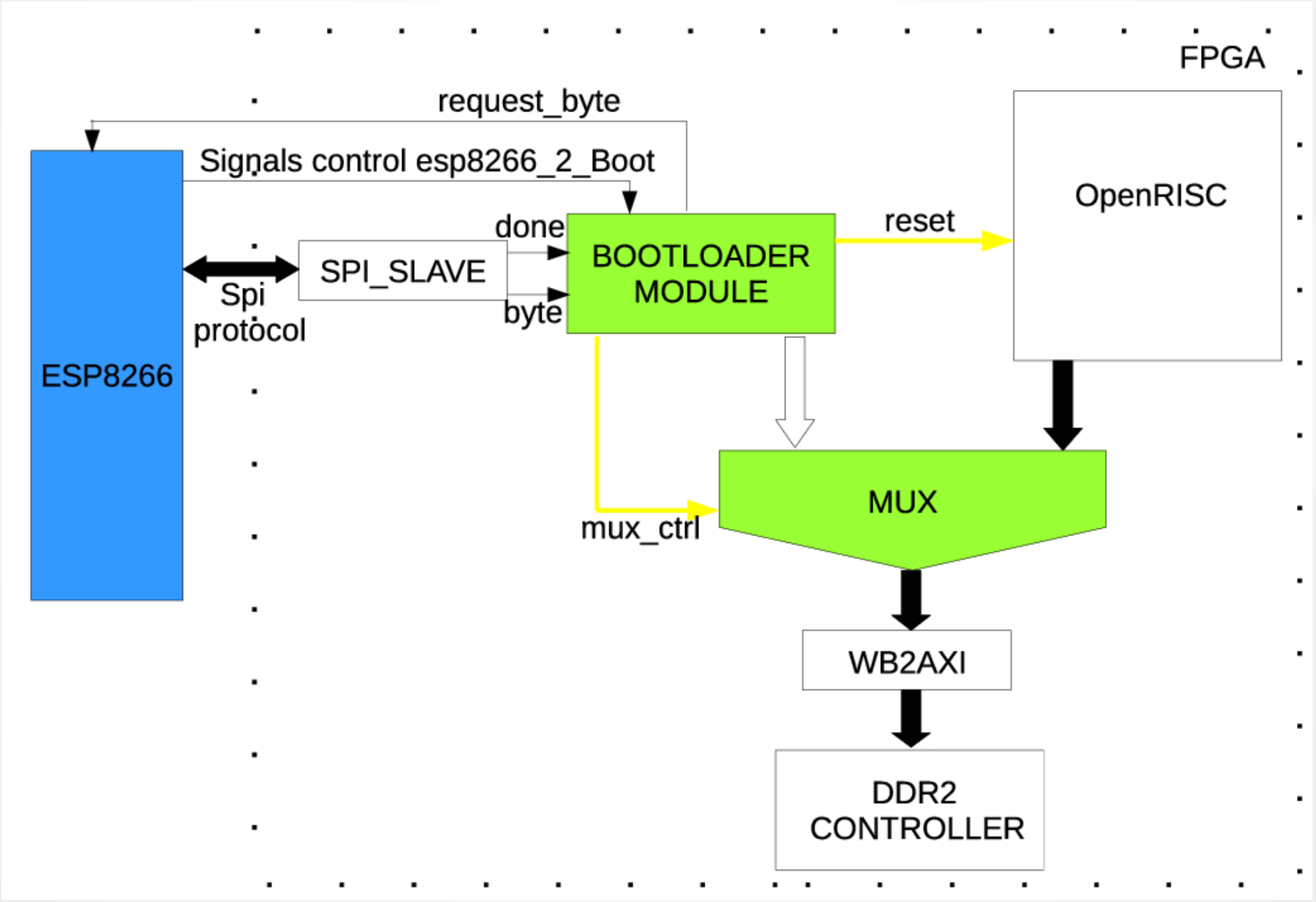
- Main Features:
  - OpenRISC(https://github.com/openrisc/orpsoc-cores/tree/master/systems/atlys)
  - DDR
    - MIG tool
    - Wb2axi - (https://github.com/wallento/wb2axi)
  - SPI slave
  - Bootloader Module
    - Designed as fusesoc core
  - nexys4ddr_nodemcu system for fusesoc

Custom Protocol

n_block	total_blocks	total_bytes	data
4 bytes	4 bytes	4 bytes	X bytes

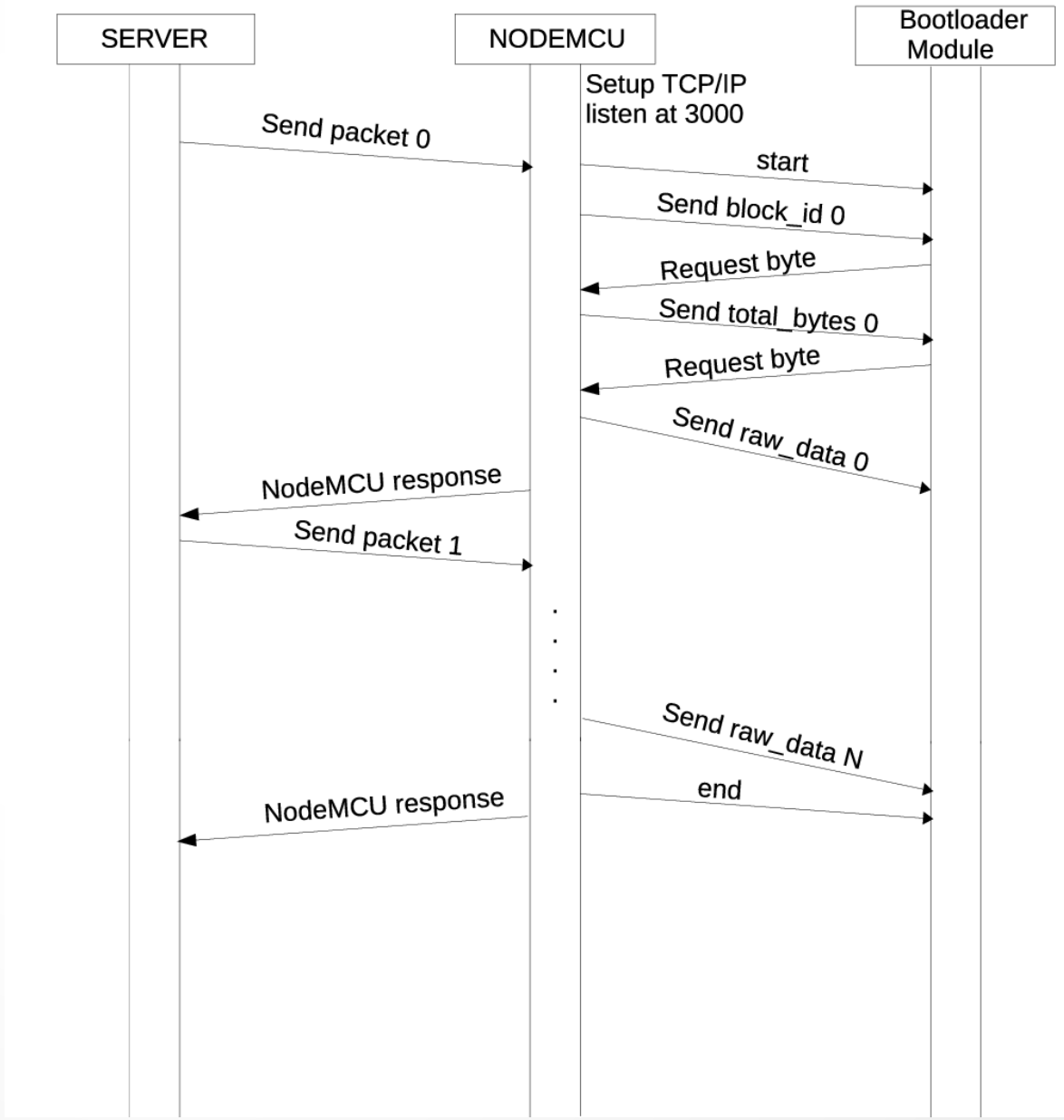
- Data packets
  - n_block : current block, 4 bytes
  - total_blocks : total number of blocks, 4 bytes
  - total_bytes : bytes in field data, 4 bytes
  - Data : chunk of binary data from the code in Base64
    - Max size of 512 bytes of the original code

SCHEMATIC


FPGA
request_byte
Signals control esp8266_2_Boot
ESP8266
Spi
protocol
SPI_SLAVE
done
byte
BOOTLOADER
MODULE
reset
OpenRISC
MUX
mux_ctrl
WB2AXI
DDR2
CONTROLLER

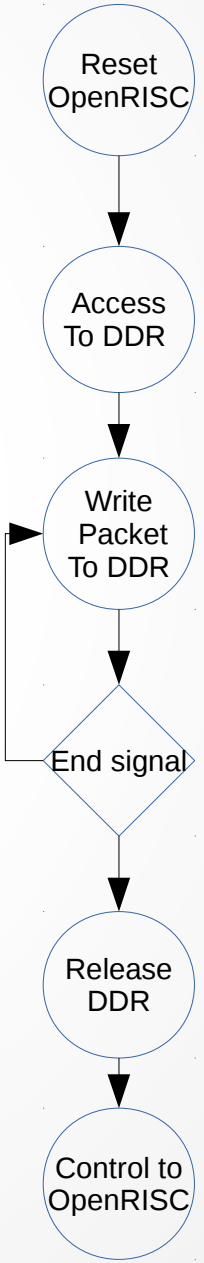
Bootloader Proccess I

- Server
  - initialize the process
- ESP8266
  - resend each packet
    to Bootloader Module
  - checks total_blocks
    - If true send end signal

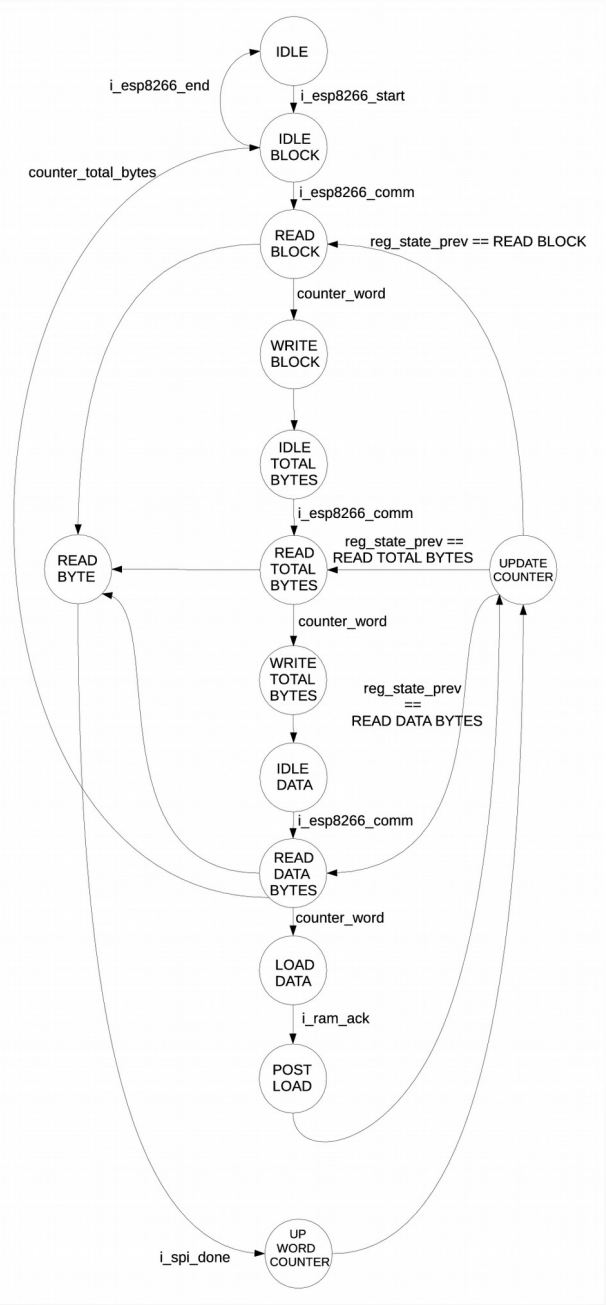


Bootloader Proccess II

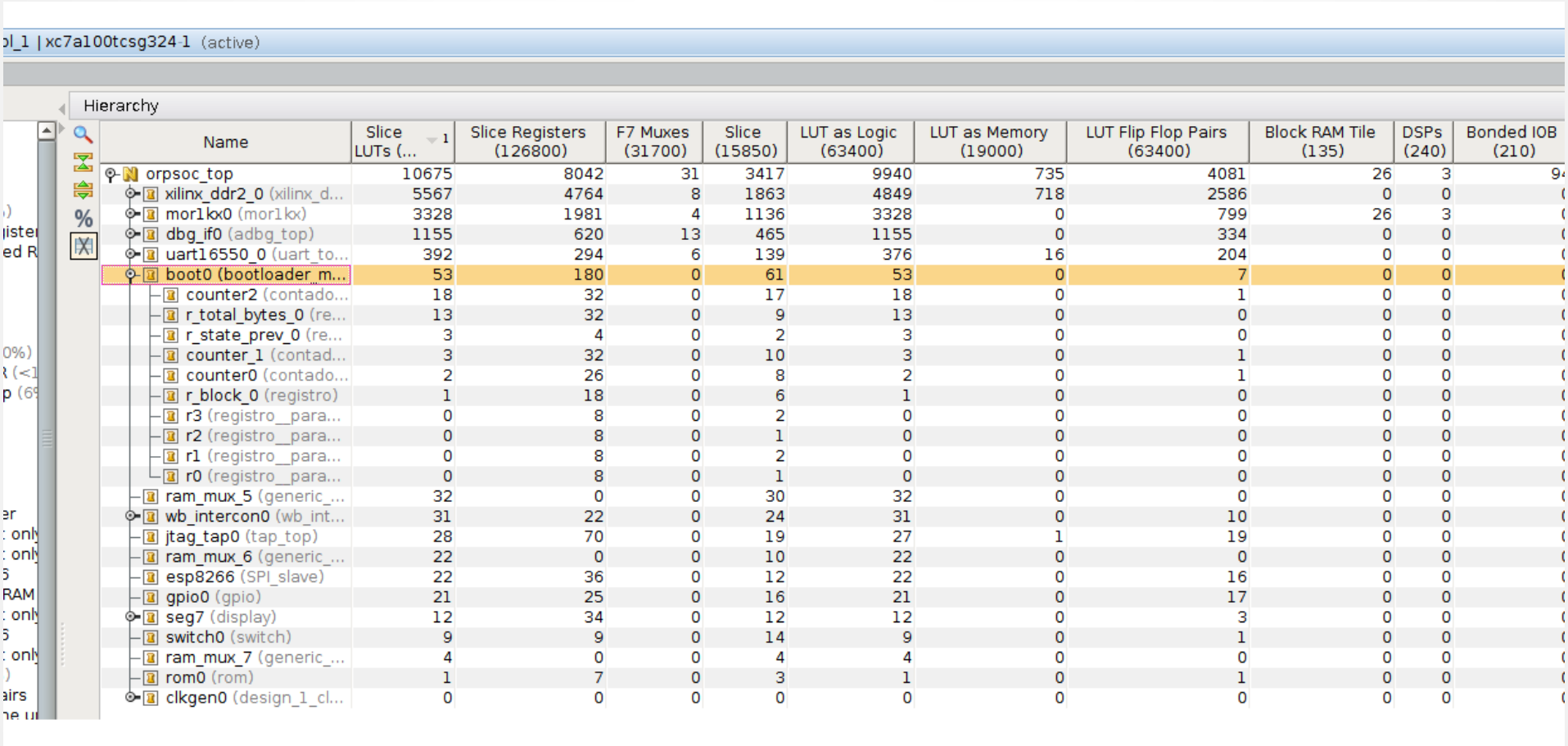
- Bootloader Module
  - Reset the OperRISC
  - Uses a MUX to get access to the DDR
  - Write receiving packet
  - Check End Signal
  - Release MUX and reset signal



Bootloader Module: State Machine



Implementations Results



ol_1 | xc7a100tcsg324-1 (active)

Hierarchy

Name	Slice LUTs (... 1	Slice Registers (126800)	F7 Muxes (31700)	Slice (15850)	LUT as Logic (63400)	LUT as Memory (19000)	LUT Flip Flop Pairs (63400)	Block RAM Tile (135)	DSPs (240)	Bonded IOB (210)
orpsoc_top	10675	8042	31	3417	9940	735	4081	26	3	9
xilinx_ddr2_0 (xilinx_d...	5567	4764	8	1863	4849	718	2586	0	0	0
mor1kx0 (mor1kx)	3328	1981	4	1136	3328	0	799	26	3	0
dbg_if0 (adbg_top)	1155	620	13	465	1155	0	334	0	0	0
uart16550_0 (uart_to...	392	294	6	139	376	16	204	0	0	0
boot0 (bootloader_m...	53	180	0	61	53	0	7	0	0	0
counter2 (contado...	18	32	0	17	18	0	1	0	0	0
r_total_bytes_0 (re...	13	32	0	9	13	0	0	0	0	0
r_state_prev_0 (re...	3	4	0	2	3	0	0	0	0	0
counter_1 (contad...	3	32	0	10	3	0	1	0	0	0
counter0 (contado...	2	26	0	8	2	0	1	0	0	0
r_block_0 (registro)	1	18	0	6	1	0	0	0	0	0
r3 (registro__para...	0	8	0	2	0	0	0	0	0	0
r2 (registro__para...	0	8	0	1	0	0	0	0	0	0
r1 (registro__para...	0	8	0	2	0	0	0	0	0	0
r0 (registro__para...	0	8	0	1	0	0	0	0	0	0
ram_mux_5 (generic_...	32	0	0	30	32	0	0	0	0	0
wb_intercon0 (wb_int...	31	22	0	24	31	0	10	0	0	0
jtag_tap0 (tap_top)	28	70	0	19	27	1	19	0	0	0
ram_mux_6 (generic_...	22	0	0	10	22	0	0	0	0	0
esp8266 (SPI_slave)	22	36	0	12	22	0	16	0	0	0
gpio0 (gpio)	21	25	0	16	21	0	17	0	0	0
seg7 (display)	12	34	0	12	12	0	3	0	0	0
switch0 (switch)	9	9	0	14	9	0	1	0	0	0
ram_mux_7 (generic_...	4	0	0	4	4	0	0	0	0	0
rom0 (rom)	1	7	0	3	1	0	1	0	0	0
clkgen0 (design_1_cl...	0	0	0	0	0	0	0	0	0	0

Slice LUTs : uses 0.49% of the total of orpsoc_top.
Slice Registers: uses 2.23% of the total of orpsoc_top.

Future Improvements

- Server
  - User Interface
- ESP8266
  - Secure channel
- SoC-OpenRISC
  - Rollback protocol

GITHUB

GITHUB PAGE

https://github.com/germancq/BOW