VLSI Circuit Design II– EHB 425E

HOMEWORK VII

Yiğit Bektaş GÜRSOY

040180063

Rana TİLKİ

040180741

Class Lecturer: Sıddıka Berna Örs Yalçın

Class Assistant:
Yasin Fırat Kula

1. Designing the Control Word

- we (Write Enable) - It is used to allow writing into the registers.
- MB - Selects the second input to the functional unit. (MuxB_sel)
- MD - Selects the data input to the Mux_D. (MuxD_sel)
- MR - Selects the data input to the Mux_Reg. (MuxR_sel)
- FS - Selects the operation of the functional unit.
- AA - Selects the 'A' input for the register file.
- BB - Selects the 'B' input for the register file.
- DA - Decides which address in the register file is selected for write operations.

DA	AA	BA	FS	MR	MB	MD	we
22-18	17-13	12-8	7-4	3	2	1	0

DA, AA, BA		FS		MR		MB		MD		we	
Function	Code	Function	Code	Function	Code	Function	Code	Function	Code	Function	Code

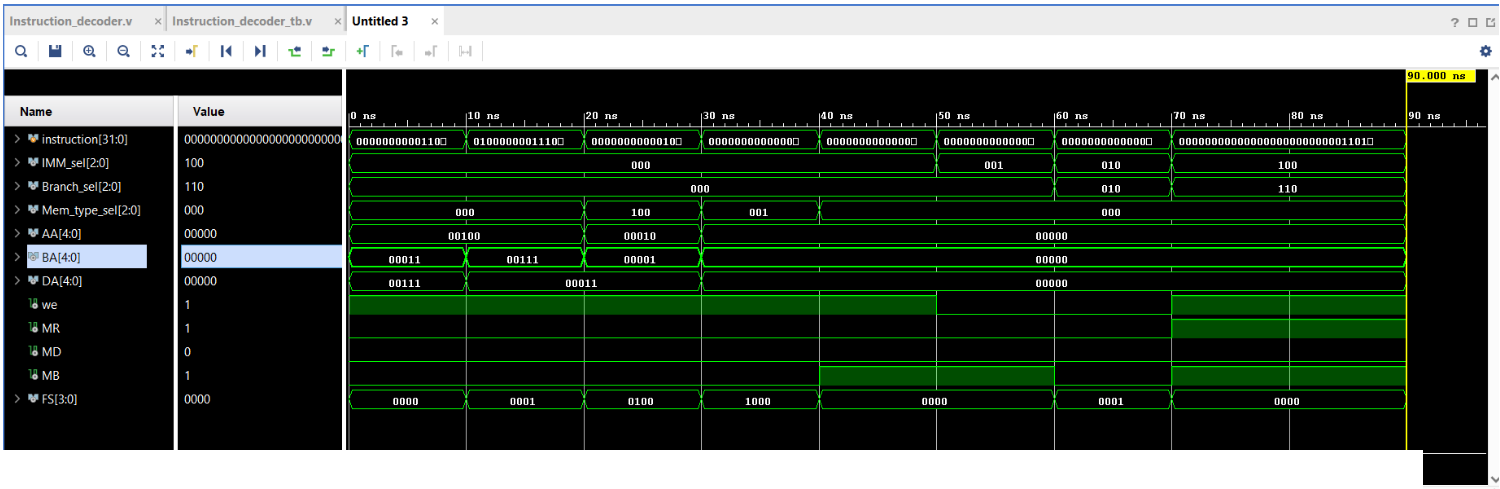
2. Instruction Decoder

1. **instruction**: This is the input instruction of a fixed size (default is 32 bits) that needs to be decoded.

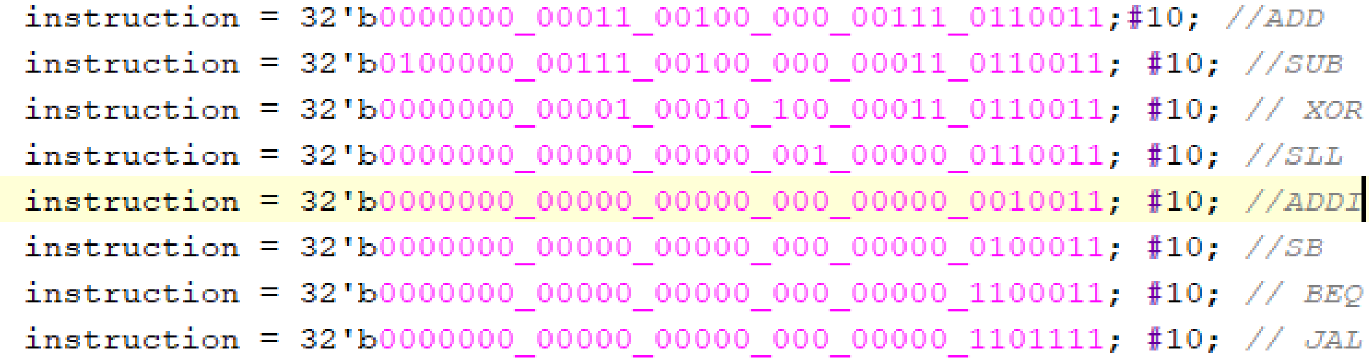
Outputs:

1. **IMM_sel**: This is a 3-bit output signal that helps determine how to select the immediate value in an instruction.
2. **Branch_sel**: This is another 3-bit output signal, used to select the branch type based on the instruction.
3. **Mem_type_sel**: This is a 3-bit output signal used to select the memory type based on the decoded instruction.
4. **AA**, **BA**, **DA**: These output signals are used to select registers based on the instruction. The width of these signals is defined by the logarithm base 2 of the instruction size, rounding up.
5. **we**: This is a write-enable signal.
6. **MR**, **MD**, **MB**: These are control signals produced by the decoder.
7. **FS**: This is a 4-bit output signal used for function selection, presumably based on the instruction type.

As can be seen in the testbench, the machine code of various operators is given. By looking at these codes, it can be decided whether the code is working correctly or not, according to whether certain signals are lit. In the first part, there is the addition operation, the we signal must be 1. And the fsel signal should be 0000 because when 0000 it is set to add and when 0001 it is set to subtraction, the other signal is subtraction, it shows correct. Then, machine code of XOR , SLL , ADDI , SB , BEQ and JAL operations are given respectively. MUXB is used in ADDI and SB transactions. The MUXB signal appears to be 1.

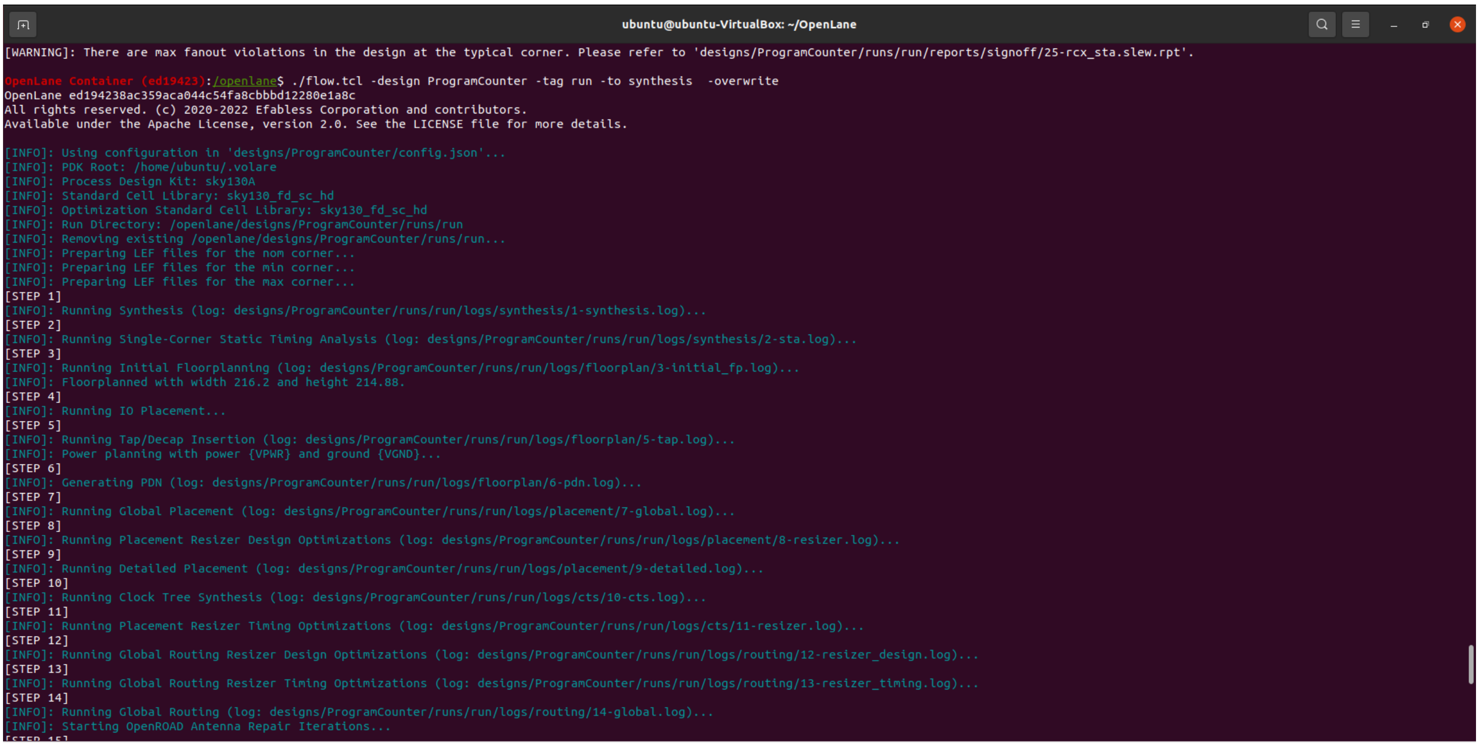


Name	Value
instruction[31:0]	0000000000000000000000000
IMM_sel[2:0]	100
Branch_sel[2:0]	110
Mem_type_sel[2:0]	000
AA[4:0]	00000
BA[4:0]	00000
DA[4:0]	00000
we	1
MR	1
MD	0
MB	1
FS[3:0]	0000



```
instruction = 32'b0000000_00011_00100_000_00111_0110011;#10; //ADD
instruction = 32'b0100000_00111_00100_000_00011_0110011; #10; //SUB
instruction = 32'b0000000_00001_00010_100_00011_0110011; #10; // XOR
instruction = 32'b0000000_00000_00000_001_00000_0110011; #10; //SLL
instruction = 32'b0000000_00000_00000_000_00000_0010011; #10; //ADDI
instruction = 32'b0000000_00000_00000_000_00000_0100011; #10; //SB
instruction = 32'b0000000_00000_00000_000_00000_1100011; #10; // BEQ
instruction = 32'b0000000_00000_00000_000_00000_1101111; #10; // JAL
```

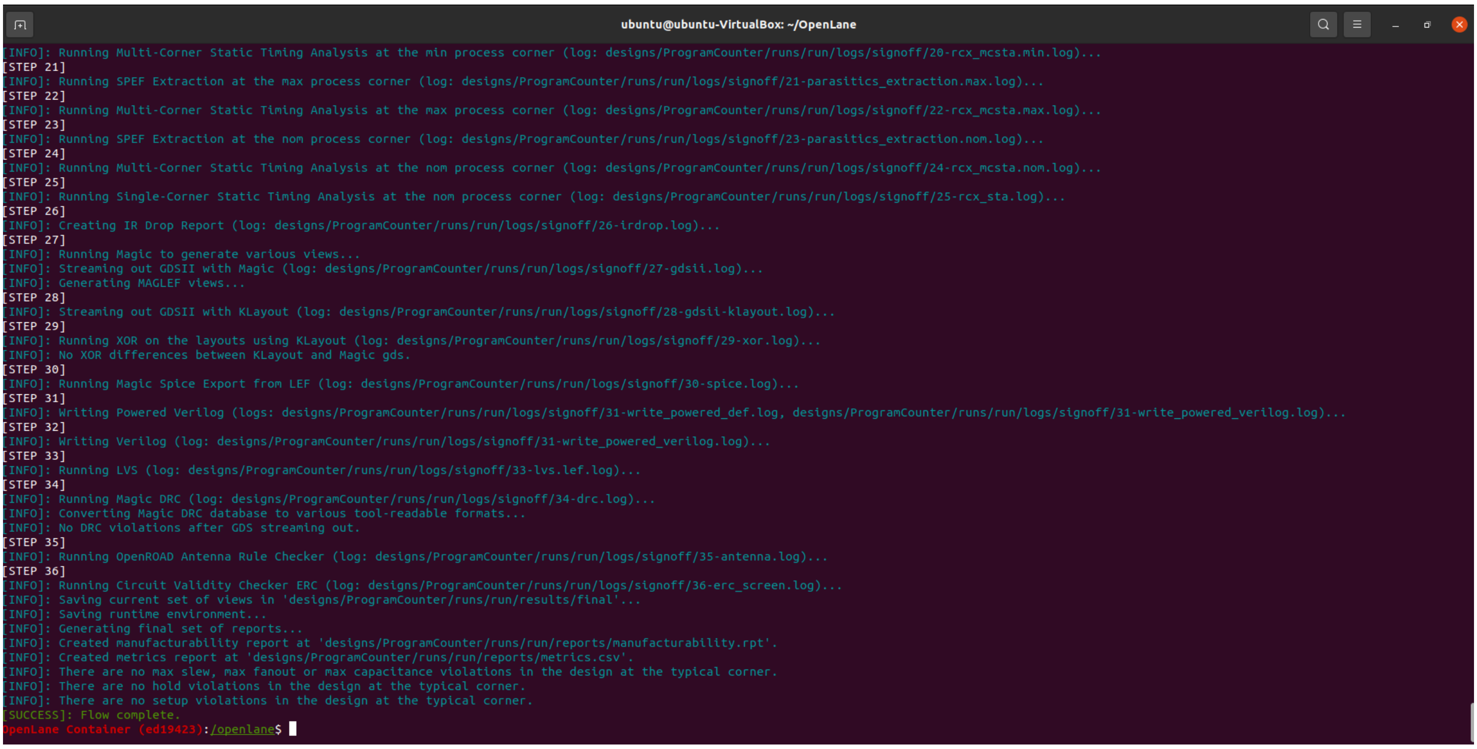
3. Program Counter



```
[WARNING]: There are max fanout violations in the design at the typical corner. Please refer to 'designs/ProgramCounter/runs/run/reports/signoff/25-rcx_sta.slew.rpt'.

OpenLane Container (ed19423):/openlane$ ./flow.tcl -design ProgramCounter -tag run -to synthesis  -overwrite
OpenLane ed194238ac359aca044c54fa8cbbbd12280e1a8c
All rights reserved. (c) 2020-2022 Efabless Corporation and contributors.
Available under the Apache License, version 2.0. See the LICENSE file for more details.

[INFO]: Using configuration in 'designs/ProgramCounter/config.json'...
[INFO]: PDK Root: /home/ubuntu/.volare
[INFO]: Process Design Kit: sky130A
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Optimization Standard Cell Library: sky130_fd_sc_hd
[INFO]: Run Directory: /openlane/designs/ProgramCounter/runs/run
[INFO]: Removing existing /openlane/designs/ProgramCounter/runs/run...
[INFO]: Preparing LEF files for the nom corner...
[INFO]: Preparing LEF files for the min corner...
[INFO]: Preparing LEF files for the max corner...
[STEP 1]
[INFO]: Running Synthesis (log: designs/ProgramCounter/runs/run/logs/synthesis/1-synthesis.log)...
[STEP 2]
[INFO]: Running Single-Corner Static Timing Analysis (log: designs/ProgramCounter/runs/run/logs/synthesis/2-sta.log)...
[STEP 3]
[INFO]: Running Initial Floorplanning (log: designs/ProgramCounter/runs/run/logs/floorplan/3-initial_fp.log)...
[INFO]: Floorplanned with width 216.2 and height 214.88.
[STEP 4]
[INFO]: Running IO Placement...
[STEP 5]
[INFO]: Running Tap/Decap Insertion (log: designs/ProgramCounter/runs/run/logs/floorplan/5-tap.log)...
[INFO]: Power planning with power {VPWR} and ground {VGND}...
[STEP 6]
[INFO]: Generating PDN (log: designs/ProgramCounter/runs/run/logs/floorplan/6-pdn.log)...
[STEP 7]
[INFO]: Running Global Placement (log: designs/ProgramCounter/runs/run/logs/placement/7-global.log)...
[STEP 8]
[INFO]: Running Placement Resizer Design Optimizations (log: designs/ProgramCounter/runs/run/logs/placement/8-resizer.log)...
[STEP 9]
[INFO]: Running Detailed Placement (log: designs/ProgramCounter/runs/run/logs/placement/9-detailed.log)...
[STEP 10]
[INFO]: Running Clock Tree Synthesis (log: designs/ProgramCounter/runs/run/logs/cts/10-cts.log)...
[STEP 11]
[INFO]: Running Placement Resizer Timing Optimizations (log: designs/ProgramCounter/runs/run/logs/cts/11-resizer.log)...
[STEP 12]
[INFO]: Running Global Routing Resizer Design Optimizations (log: designs/ProgramCounter/runs/run/logs/routing/12-resizer_design.log)...
[STEP 13]
[INFO]: Running Global Routing Resizer Timing Optimizations (log: designs/ProgramCounter/runs/run/logs/routing/13-resizer_timing.log)...
[STEP 14]
[INFO]: Running Global Routing (log: designs/ProgramCounter/runs/run/logs/routing/14-global.log)...
[INFO]: Starting OpenROAD Antenna Repair Iterations...
[STEP 15]
```



```
[INFO]: Running Multi-Corner Static Timing Analysis at the min process corner (log: designs/ProgramCounter/runs/run/logs/signoff/20-rcx_mcsta.min.log)...
[STEP 21]
[INFO]: Running SPEF Extraction at the max process corner (log: designs/ProgramCounter/runs/run/logs/signoff/21-parasitics_extraction.max.log)...
[STEP 22]
[INFO]: Running Multi-Corner Static Timing Analysis at the max process corner (log: designs/ProgramCounter/runs/run/logs/signoff/22-rcx_mcsta.max.log)...
[STEP 23]
[INFO]: Running SPEF Extraction at the nom process corner (log: designs/ProgramCounter/runs/run/logs/signoff/23-parasitics_extraction.nom.log)...
[STEP 24]
[INFO]: Running Multi-Corner Static Timing Analysis at the nom process corner (log: designs/ProgramCounter/runs/run/logs/signoff/24-rcx_mcsta.nom.log)...
[STEP 25]
[INFO]: Running Single-Corner Static Timing Analysis at the nom process corner (log: designs/ProgramCounter/runs/run/logs/signoff/25-rcx_sta.log)...
[STEP 26]
[INFO]: Creating IR Drop Report (log: designs/ProgramCounter/runs/run/logs/signoff/26-irdrop.log)...
[STEP 27]
[INFO]: Running Magic to generate various views...
[INFO]: Streaming out GDSII with Magic (log: designs/ProgramCounter/runs/run/logs/signoff/27-gdsii.log)...
[INFO]: Generating MAGLEF views...
[STEP 28]
[INFO]: Streaming out GDSII with KLayout (log: designs/ProgramCounter/runs/run/logs/signoff/28-gdsii-klayout.log)...
[STEP 29]
[INFO]: Running XOR on the layouts using KLayout (log: designs/ProgramCounter/runs/run/logs/signoff/29-xor.log)...
[INFO]: No XOR differences between KLayout and Magic gds.
[STEP 30]
[INFO]: Running Magic Spice Export from LEF (log: designs/ProgramCounter/runs/run/logs/signoff/30-spice.log)...
[STEP 31]
[INFO]: Writing Powered Verilog (logs: designs/ProgramCounter/runs/run/logs/signoff/31-write_powered_def.log, designs/ProgramCounter/runs/run/logs/signoff/31-write_powered_verilog.log)...
[STEP 32]
[INFO]: Writing Verilog (log: designs/ProgramCounter/runs/run/logs/signoff/31-write_powered_verilog.log)...
[STEP 33]
[INFO]: Running LVS (log: designs/ProgramCounter/runs/run/logs/signoff/33-lvs.lef.log)...
[STEP 34]
[INFO]: Running Magic DRC (log: designs/ProgramCounter/runs/run/logs/signoff/34-drc.log)...
[INFO]: Converting Magic DRC database to various tool-readable formats...
[INFO]: No DRC violations after GDS streaming out.
[STEP 35]
[INFO]: Running OpenROAD Antenna Rule Checker (log: designs/ProgramCounter/runs/run/logs/signoff/35-antenna.log)...
[STEP 36]
[INFO]: Running Circuit Validity Checker ERC (log: designs/ProgramCounter/runs/run/logs/signoff/36-erc_screen.log)...
[INFO]: Saving current set of views in 'designs/ProgramCounter/runs/run/results/final'...
[INFO]: Saving runtime environment...
[INFO]: Generating final set of reports...
[INFO]: Created manufacturability report at 'designs/ProgramCounter/runs/run/reports/manufacturability.rpt'.
[INFO]: Created metrics report at 'designs/ProgramCounter/runs/run/reports/metrics.csv'.
[INFO]: There are no max slew, max fanout or max capacitance violations in the design at the typical corner.
[INFO]: There are no hold violations in the design at the typical corner.
[INFO]: There are no setup violations in the design at the typical corner.
[SUCCESS]: Flow complete.
OpenLane Container (ed19423):/openlane$
```