



CS250 - Computer Organization and Architecture

nPower Processor in TL-Verilog - HDL Assignment

28.05.2021

Team members:

- 1) Himanshu Kumar - 191CS122
- 2) Praneeth G - 191CS235
- 3) Amal Majunu Vidya - 191CS107
- 4) Aniket Srivastava - 191CS208

Content

Content	1
64-bit Integer ALU Design	2
Block Diagram	2
ALU Control	3
Block Diagram	4
Instruction Fetch Hardware	5
Block Diagram	5
Datapath	6
add instruction	6
addi instruction	7
and instruction	8
beq instruction	9
load instruction	10
or instruction	11
store instruction	12
sub instruction	13
nPower Processor Pipeline	14
Block Diagram	14
THANK YOU!	15

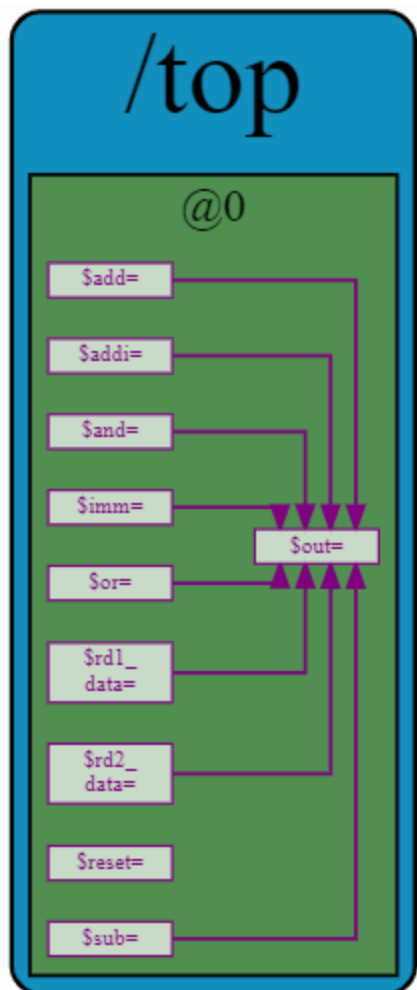
64-bit Integer ALU Design

The 64-bit ALU performs can perform following operations to two 64-bit operands:

1. and : &&
2. or : ||
3. addition : +
4. subtraction : -

This 64-bit ALU requires the output from the ALU control unit as input which helps in identifying the operation to be done between the two operands.

Block Diagram



ALU Control

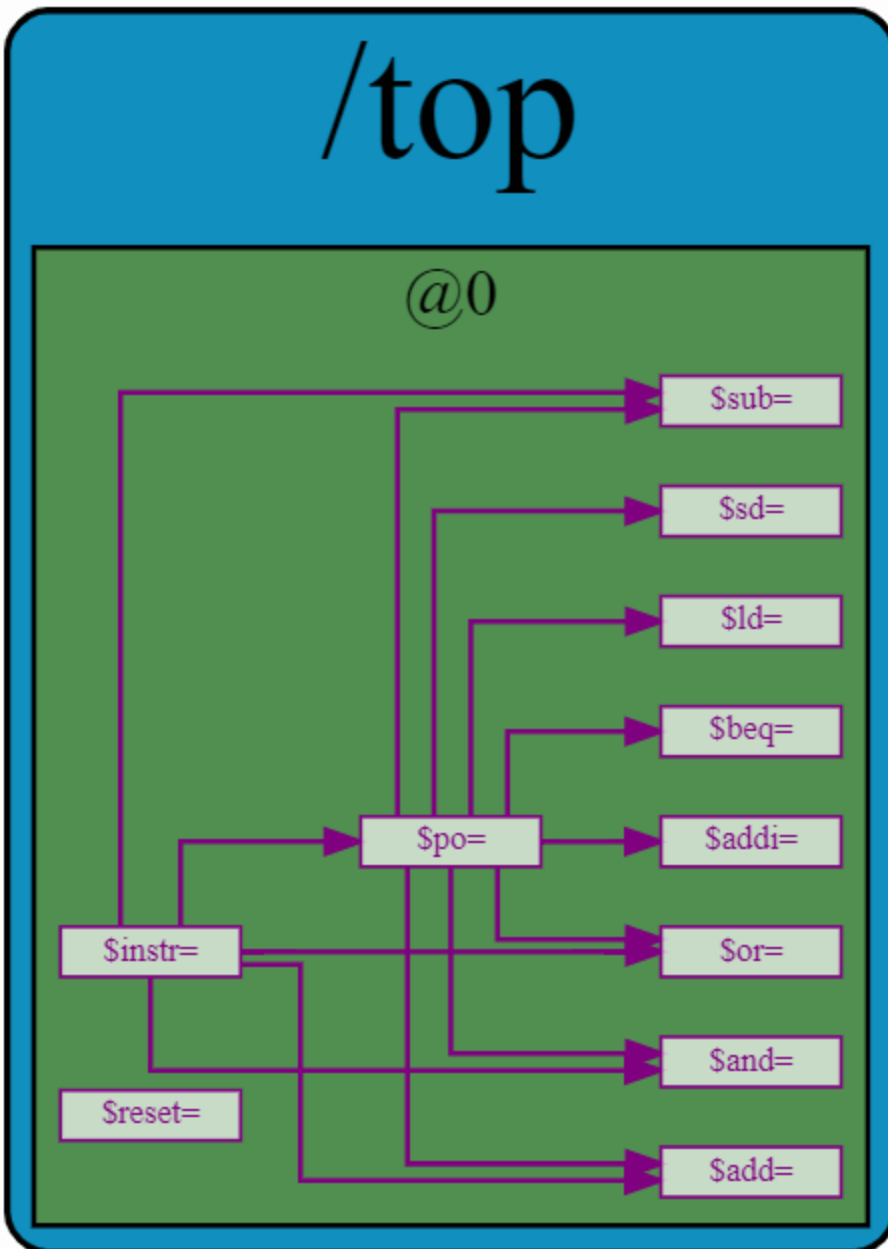
As the name suggests it controls the ALU designed above.

It informs the operation to be done by the ALU for the corresponding instruction.

Following table shows the ALU operation for the corresponding instruction:

Instruction	Operation by ALU
add	addition
addi	addition
sub	subtraction
ld	addition
sd	addition
and	bitwise and
or	bitwise or

Block Diagram



Instruction Fetch Hardware

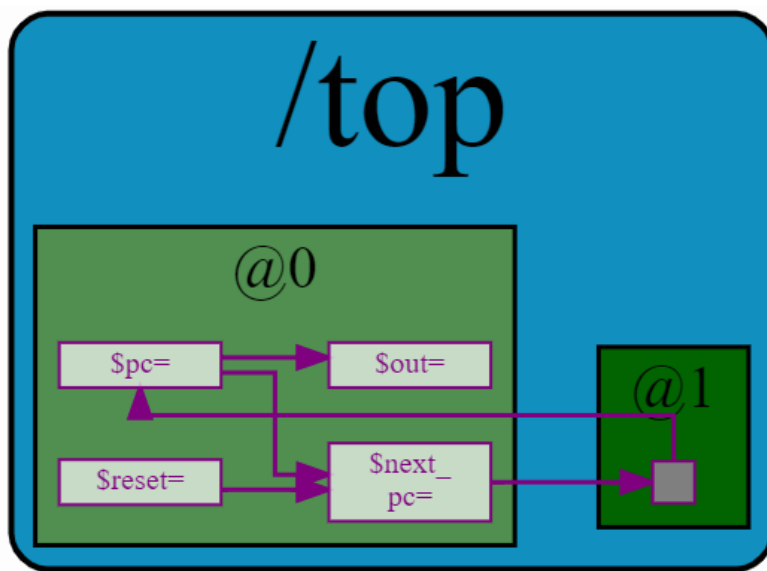
As the name suggests it reads the instruction from the memory.

Based on the value of PC it reads/fetches the corresponding instruction in memory at PCth location in the memory.

It updates PC value to PC + 4 as default.

If the reset value is set to 1 then it resets the PC value such that it reads the 1st instruction.

Block Diagram



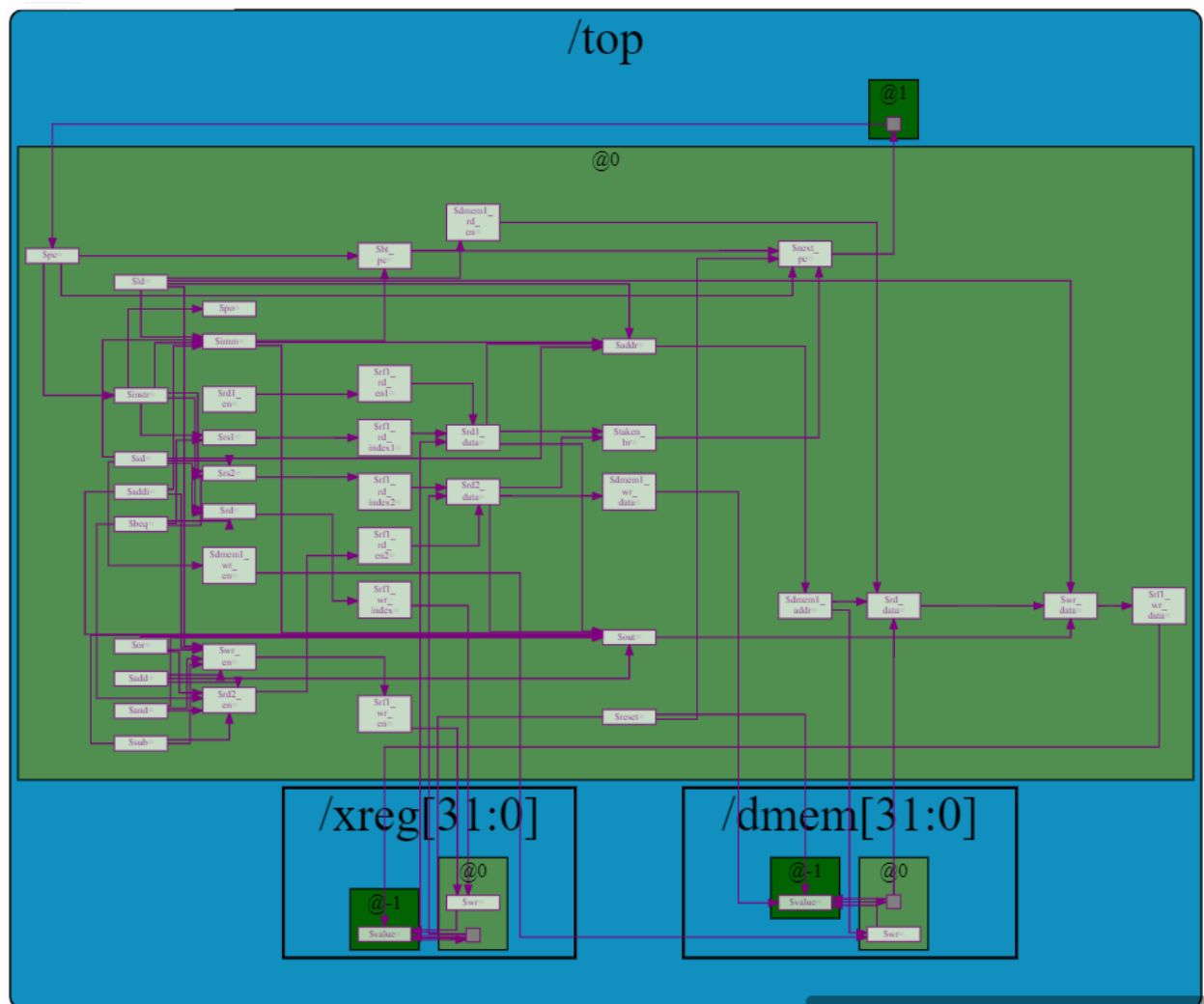
Datapath

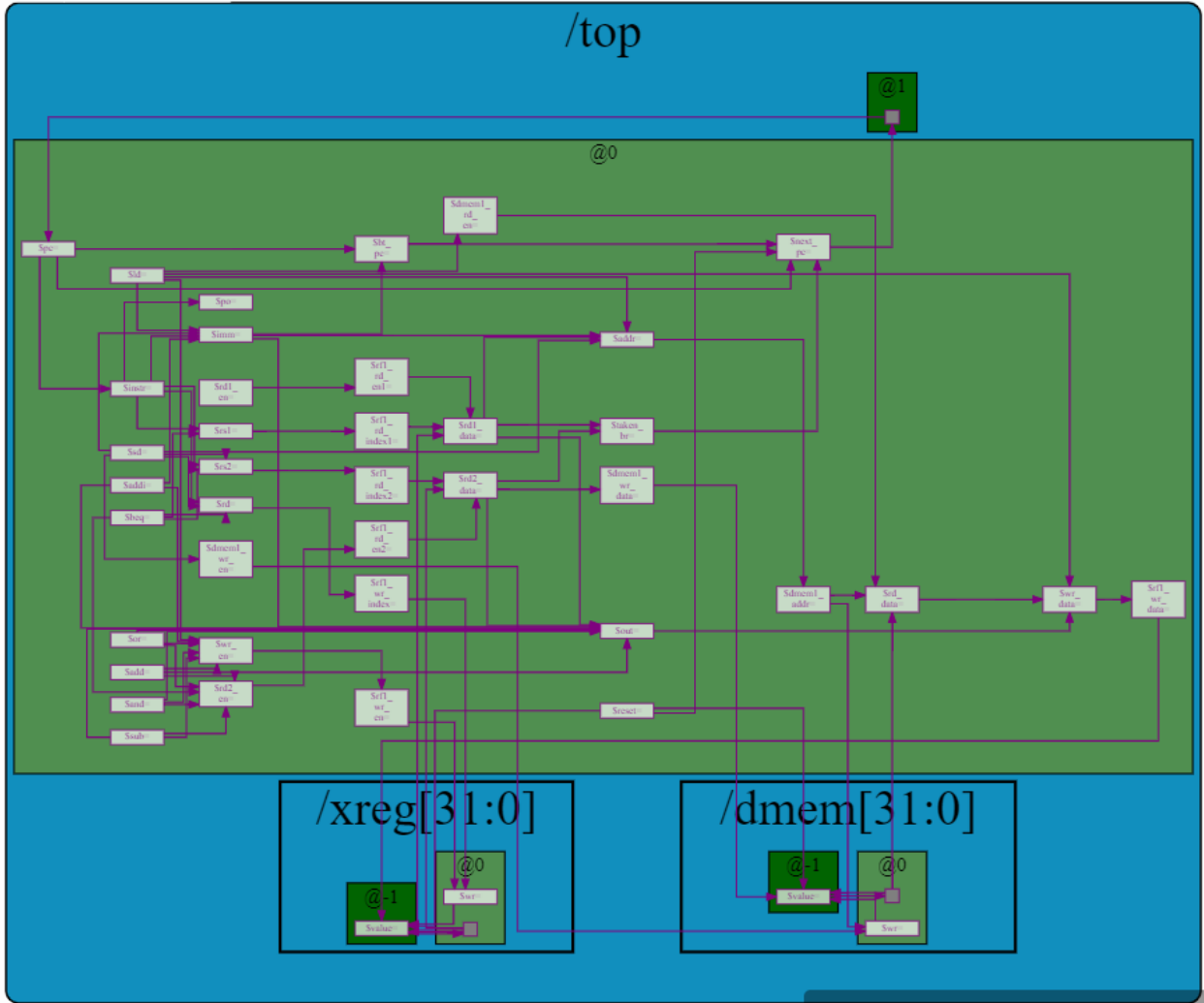
Now this code can execute a complete instruction.

Here the control signals are hard coded accordingly for a particular instruction to get executed.

Now the block diagrams for each instruction is as follows:

add instruction





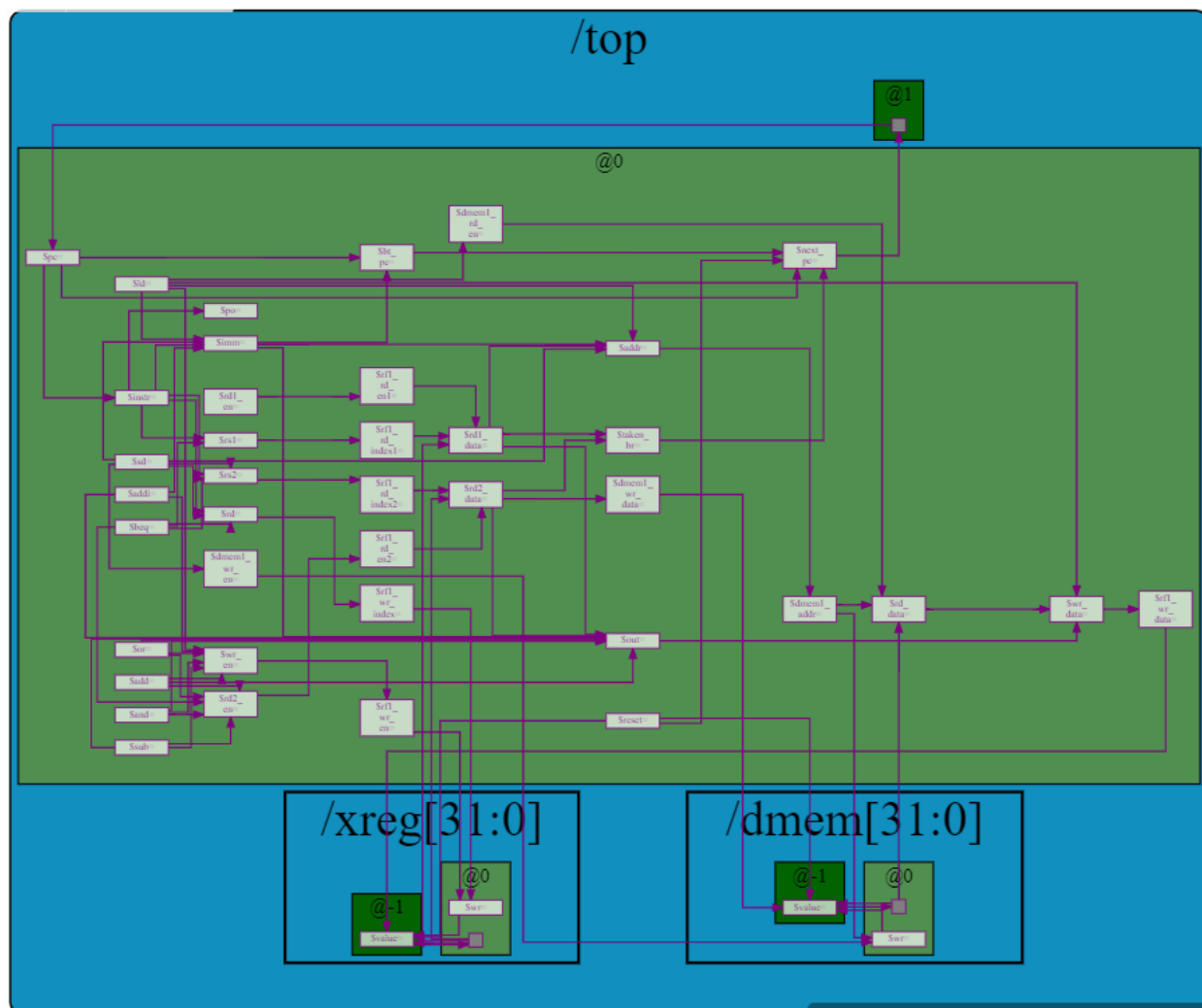
[illegible]

The diagram illustrates a complex processor core, labeled **@0**, which is part of a system with a top-level block **@1**. The core's internal components include:

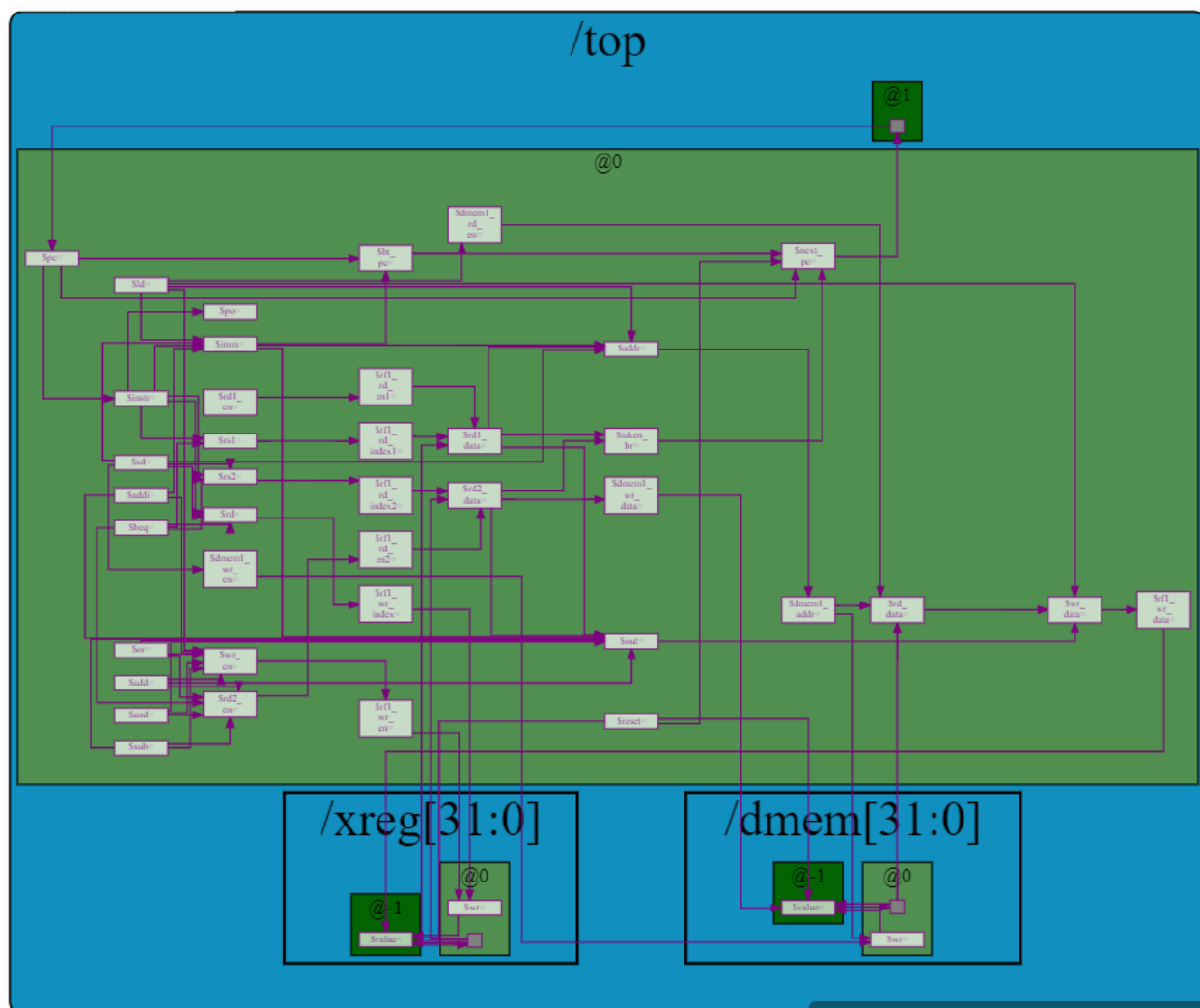
- Registers:** A large array of registers on the left, including **\$pc**, **\$sp**, **\$sra**, **\$srd**, **\$srd2**, **\$srd3**, **\$srd4**, **\$srd5**, **\$srd6**, **\$srd7**, **\$srd8**, **\$srd9**, **\$srd10**, **\$srd11**, **\$srd12**, **\$srd13**, **\$srd14**, **\$srd15**, **\$srd16**, **\$srd17**, **\$srd18**, **\$srd19**, **\$srd20**, **\$srd21**, **\$srd22**, **\$srd23**, **\$srd24**, **\$srd25**, **\$srd26**, **\$srd27**, **\$srd28**, **\$srd29**, **\$srd30**, **\$srd31**.
- ALU and FPU:** A central ALU block and a floating-point unit (FPU) block on the right, both connected to the register file.
- Cache and Memory:** A cache block at the bottom, connected to the register file and the FPU. It interfaces with external memory blocks labeled **/xreg[31:0]** and **/dmem[31:0]**.
- Control and Status:** A control block at the top, connected to the register file and the FPU.

The core is connected to a top-level block **@1** and interfaces with external memory blocks **/xreg[31:0]** and **/dmem[31:0]**.

store instruction



sub instruction

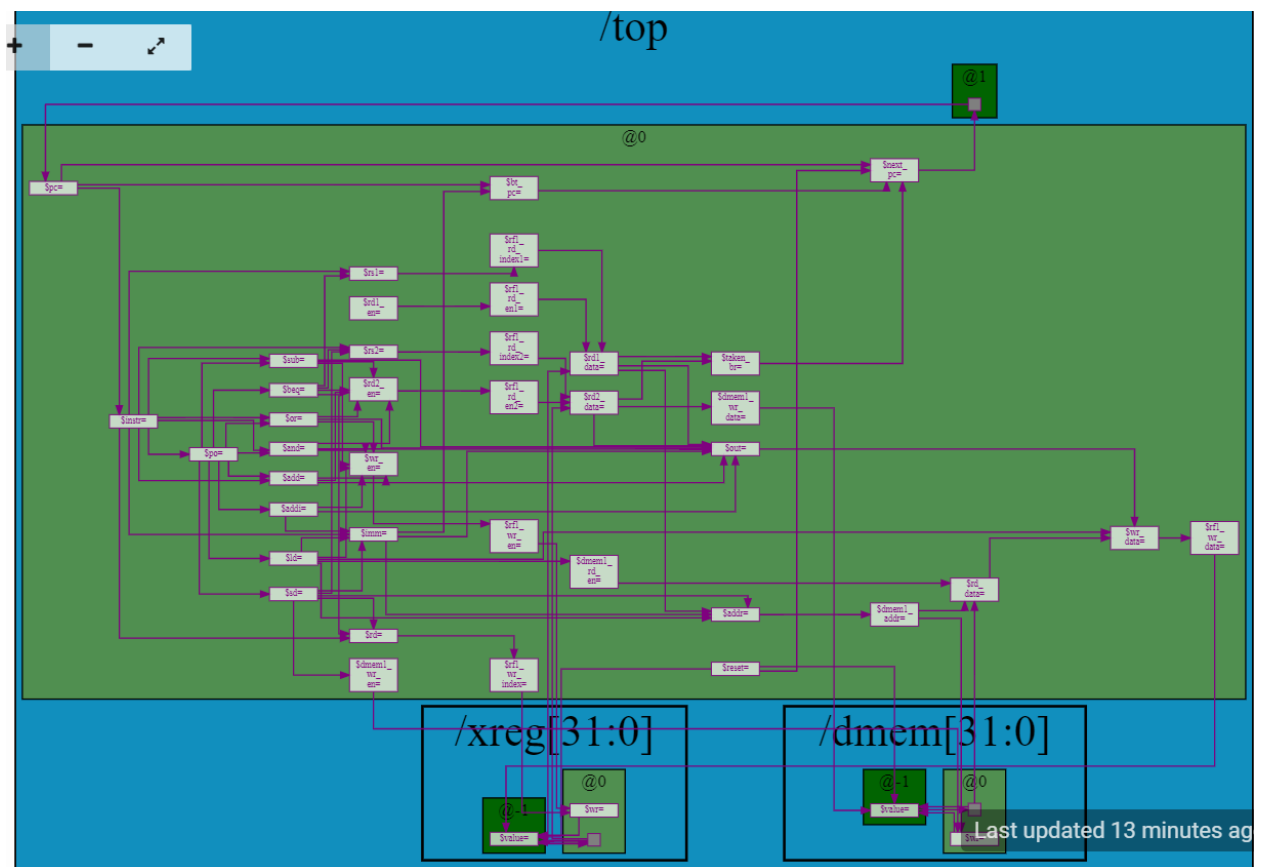


nPower Processor Pipeline

The pipeline is obtained by integrating all the above created modules and including a main control unit instead of hard coding control signals.

Here at every cycle, according to the program counter, the instructions get executed in a pipeline manner.

Block Diagram





THANK YOU!