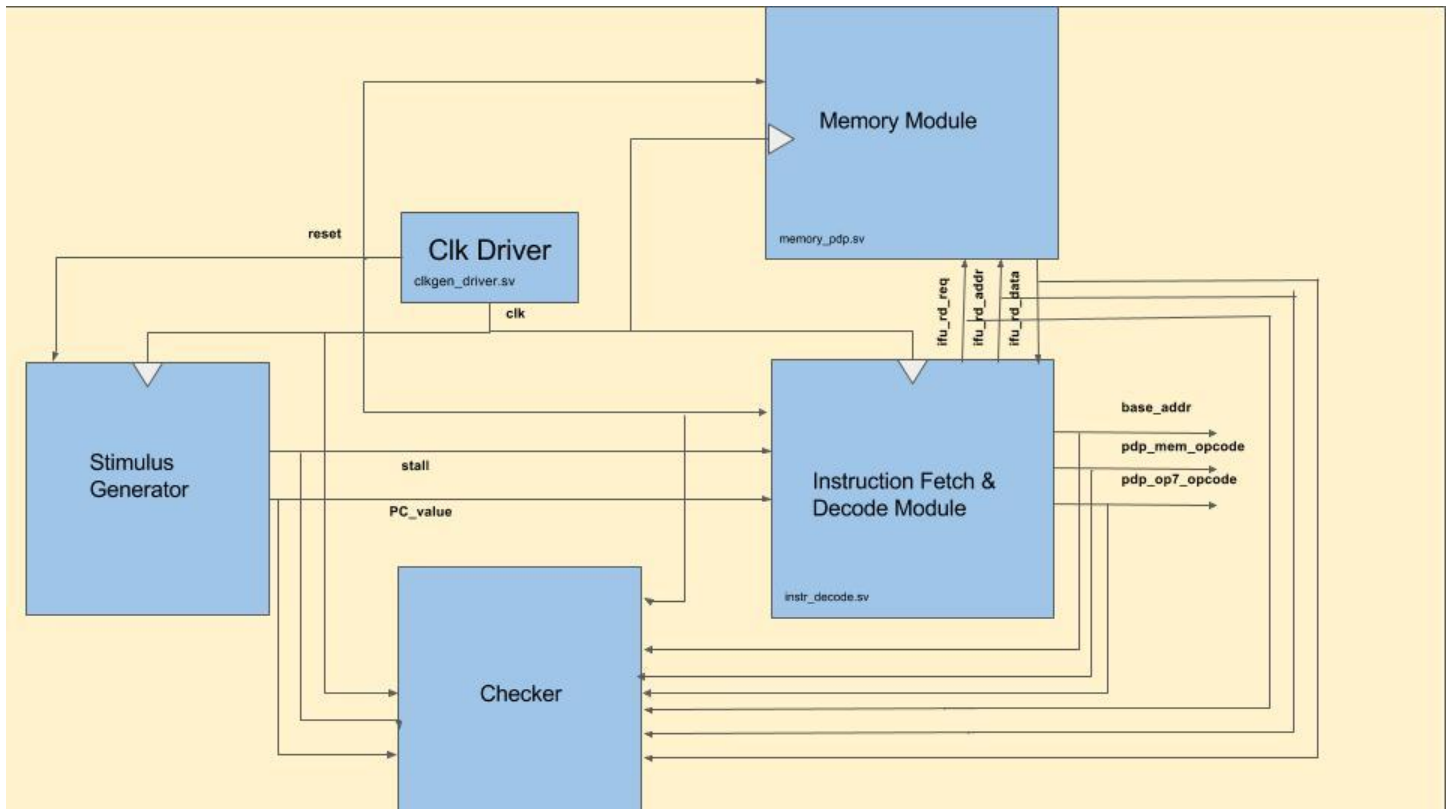


ECE 593 / Final Project – Proposal: Team ‘P’

Verification Plan: Hardware implementation of PDP8 Instruction Set Architecture (ISA) level simulator

1. Unit level testing for Instruction Fetch and Decode block:



Checking strategy: Grey box approach for state transitions in compliment with black box approach for input/output checking.

Components:

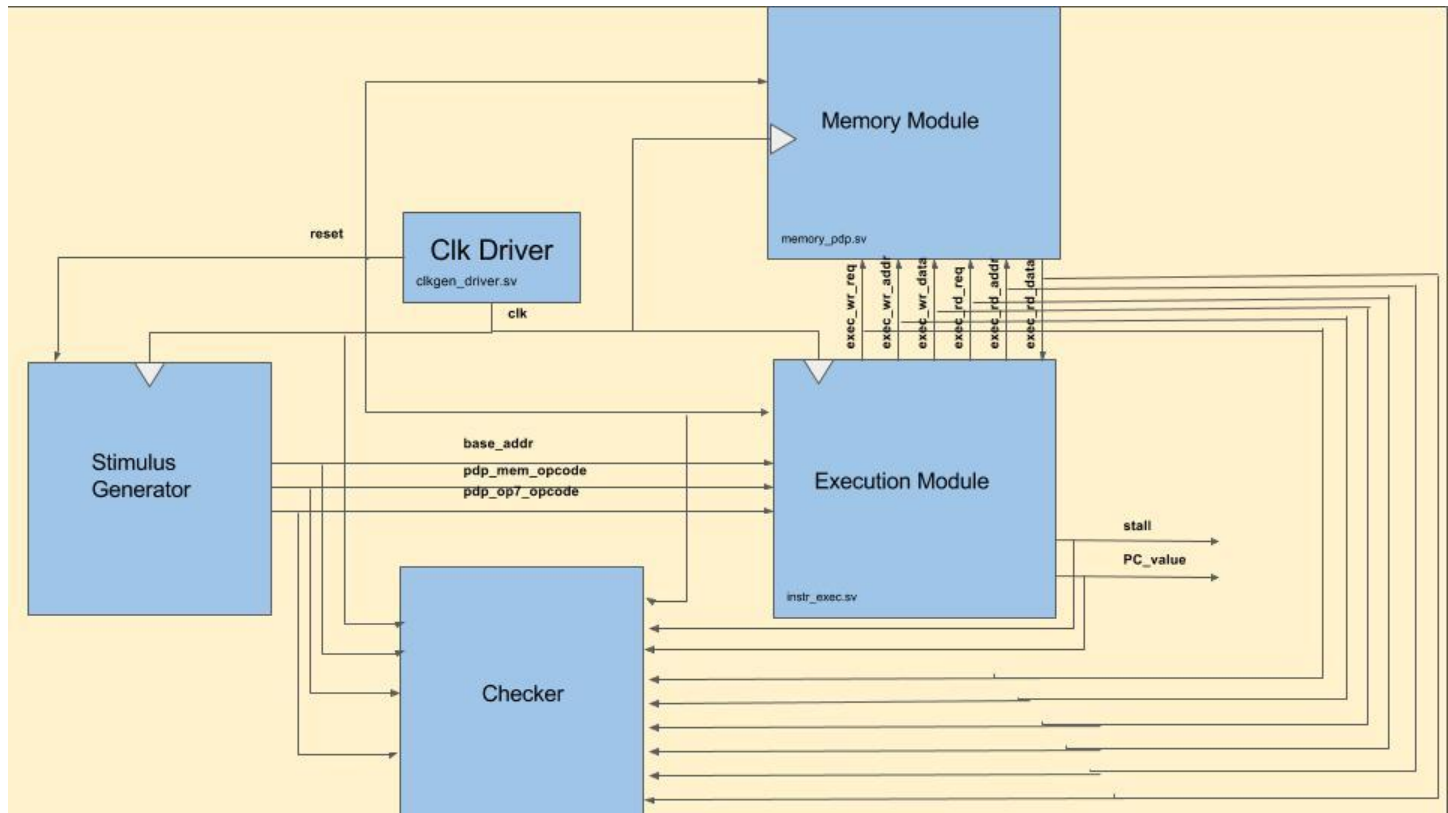
1. **Stimulus Generator:** Mimics Execution unit in giving inputs to Instruction Fetch and Decode module.
2. **Clk Driver:** Clock generator with initial reset and pre-defined runtime.
3. **Checker:** To check for correct outputs and state transitions.
4. **Instruction Fetch and Decode Module:** Design under test.
5. **Memory module:** To fetch instructions from memory and decode.

Functions to be verified using assertions:

1. State transitions based on the inputs to DUT.
2. Reset functionality.

- Valid base address.
- Clearing outputs pdp_mem_opcode, pdp_op7_opcode in specific states mentioned.
- Assertion and de-assertion of read request in specific states mentioned.
- Conversion of instructions fetched from memory to correct opcodes and memory instruction addresses.
- Correct update of PC value when out of stall.
- Termination of the execution when PC value equals to pre-defined base address.

2. Unit level testing for Execute block:



Checking strategy: Grey box approach for state transitions in compliment with black box approach for input/output checking.

Components:

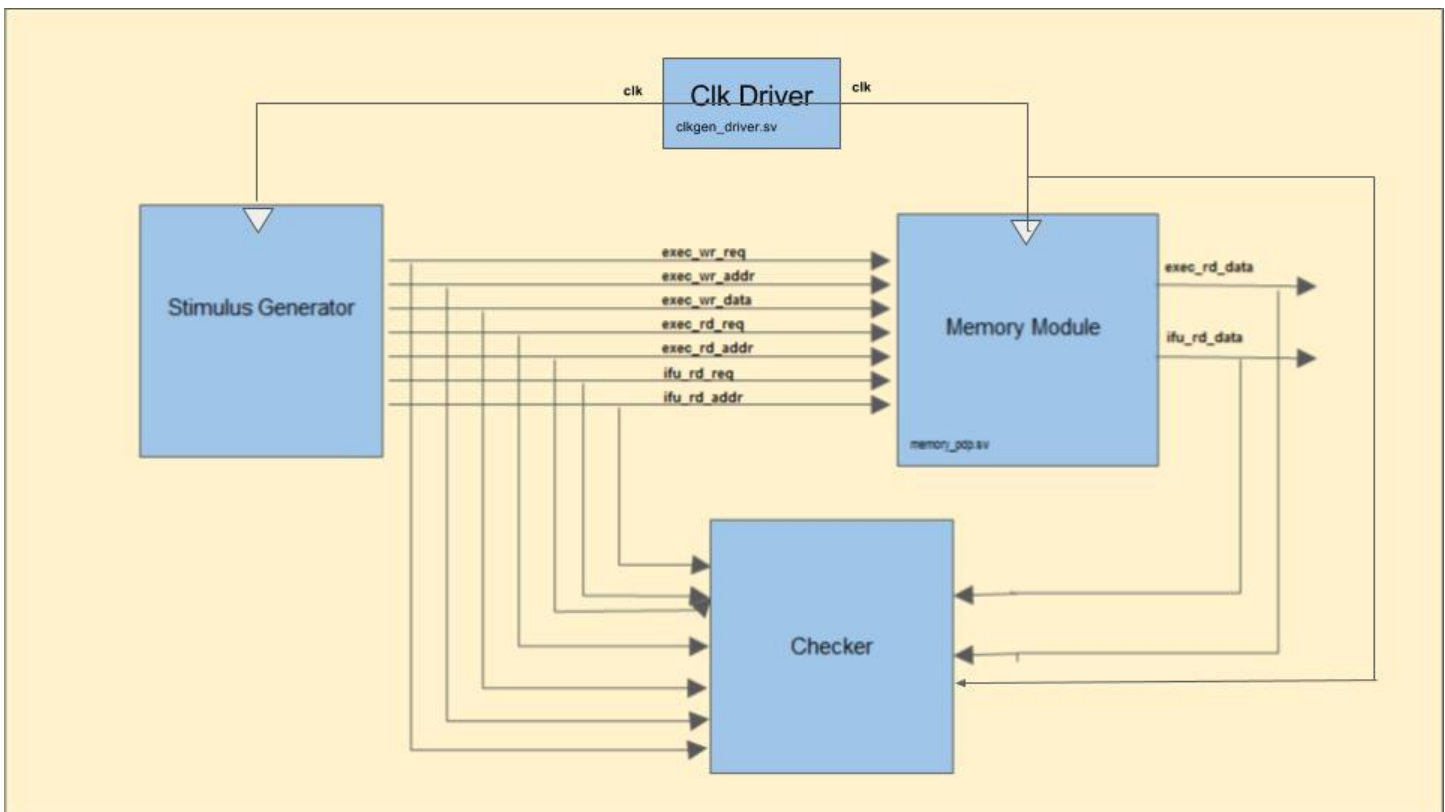
- Stimulus Generator:** Mimics Instruction Fetch and Decode unit in giving inputs to Execution module.
- Clk Driver:** Clock generator with initial reset and pre-defined runtime.
- Checker:** To check for correct outputs and state transitions.
- Execution Module:** Design under test.
- Memory module:** Execution block processes the instruction and reads/writes from/to memory (if needed based on the instruction).

Functions to be verified using assertions:

- State transitions based on the inputs to DUT.
- Reset functionality.

3. Update of PC with base address when out of reset.
4. Assertion and de-assertion of stall signal to Instruction Fetch and Decode unit.
5. Incrementing and updating PC upon new opcode.
6. Clearing Accumulator and Link registers in specific states mentioned.
7. Assertion and de-assertion of read request in specific states mentioned.
8. Assertion and de-assertion of write request in specific states mentioned.
9. Correct update of Accumulator and link registers in specific states mentioned.
10. Correct update of memory instruction address to either read address or write address in specific states mentioned.
11. Correct update of write data with either (read data+1) or Accumulator register value or PC value in specific states mentioned.
12. Saving PC value to temporary location after issuing the new PC value to Instruction Fetch and Decode unit.

3. Unit level testing for Memory block:



Checking strategy: Grey box approach to retrieve internal data from memory for comparison of read/write, in compliment with black box approach for input/output checking for read data.

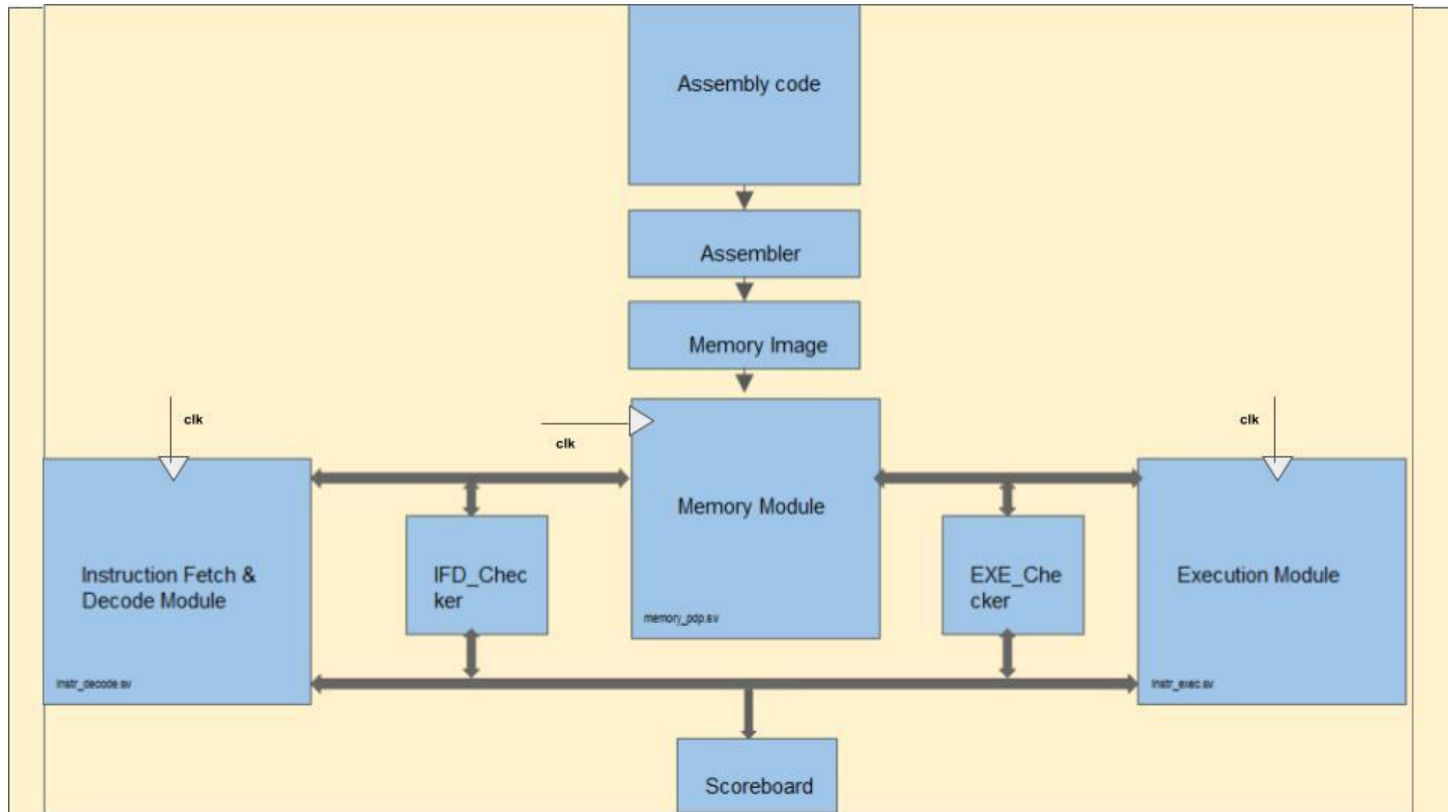
Components:

1. **Stimulus Generator:** Mimics Instruction Fetch and Decode unit and Execution unit in giving inputs to memory module.
2. **Clk Driver:** Clock generator with initial reset and pre-defined runtime.
3. **Checker:** To check for correct read/write data.
4. **Memory Module:** Design under test.

Functions to be verified using assertions:

1. Successive read after reads, write after writes, read after writes and write after reads for various locations and same location.
2. Valid memory addressing.
3. No parallel read requests from Instruction Fetch and Decode unit and Execution unit.

4. Chip level testing for the whole design:



Checking strategy: Black box approach for input/output checking.

Components:

1. **Assembly code:** Assembly test for simple arithmetic tests to PDP8 processor.
2. Clk Driver, Instruction Fetch and Decode unit, Execution unit and Memory unit.
3. **Checker:** Checkers at unit level for Instruction Fetch and Decode unit and Execution unit are bounded at chip level testing.
4. **Scoreboard:** To keep track of the instructions being executed and states entered

Testing:

Memory image from Assembly test as inputs. (Deterministic testing)

Coverage:

- Assertion based coverage for state transitions of both units.
- Coverage point at the input interface i.e; between Stimulus and DUT.