Module: R5: RV-fpga
Section: RISC-V VeeR Core **Task:** Hardware Counters

Hardware Counters

➢ **Analyzing SweRV EH1:**

Objective
The task involved analyzing a RISC-V program to understand the configuration and use of hardware performance counters in the SweRV EH1 core.

Program Overview
The program consists of two parts:

Main Function:

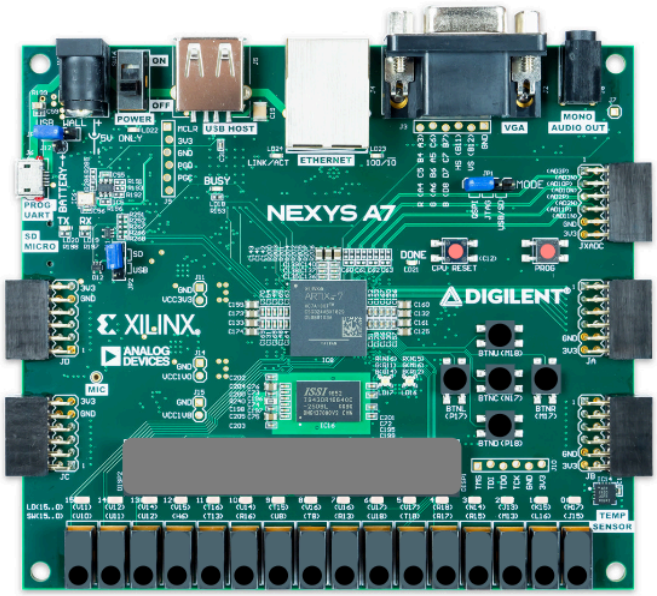
- **UART Initialization:** Sets up communication via UART.
- **Performance Counter Setup:** Configures four counters to monitor clock cycles, instructions committed, branches committed, and branch mispredictions.
- **Data Collection:** Captures and prints the counter values before and after running an assembly routine.
- **Test_Assembly()**
- **Computation Loop:** Executes arithmetic and logical operations in a loop, generating events for the counters to track.

Running on RVfpga-ViDBo:

```
                Cycles = 3000371
Instructions = 6000058
BrCom = 1000024
BrMis = 18
```

Disconnect | Clear UART output

7 SEGMENT DISPLAYS: 0 0 0 0 0 0 0 0



Summary

The program demonstrates the use of SweRV EH1 hardware counters to monitor performance metrics during execution. The counters are configured, data is collected before and after execution, and results are printed, showing cycles, instructions, branch commits, and mispredictions.

➢ **Core Features:**

- **Disabling Dual-Issue Execution:**
  For that purpose, include at the beginning of the assembly program the following instructions:

```
li t2, 0x400
```

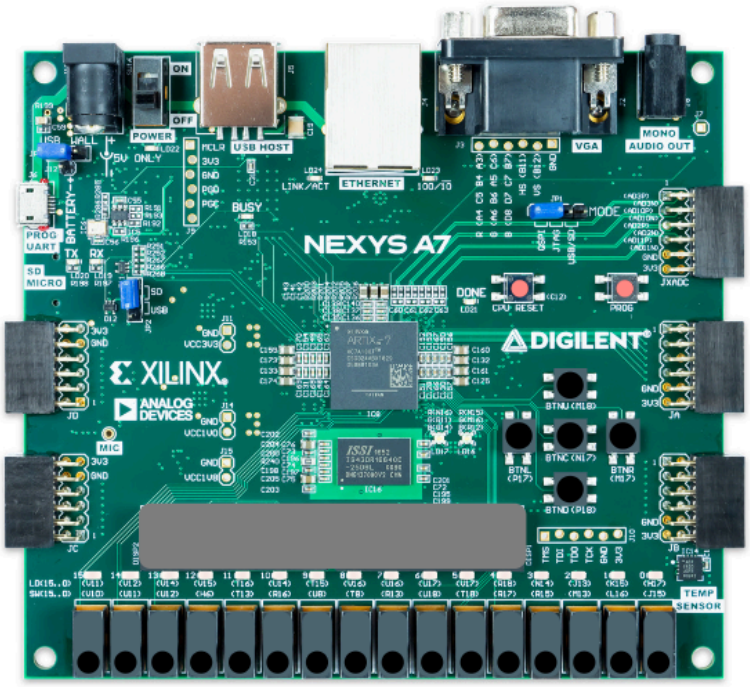
```
csrrs t1, 0x7F9, t2
```

Running on RVfpga-ViDBo:

```
Cycles = 6000401
Instructions = 6000064
BrCom = 1000026
BrMis = 19
```

Disconnect | Clear UART output

7 SEGMENT DISPLAYS: 0 0 0 0 0 0 0 0



Summary

Disabling dual-issue simplifies the processor's execution model, but it sacrifices the ability to exploit instruction-level parallelism (ILP). In a scenario where dual-issue execution is possible, it can significantly improve performance by executing more instructions in fewer cycles. Disabling it forces the processor to handle instructions sequentially, leading to higher cycle counts and potentially

slower overall performance. The performance counters reflect this change, primarily through an increase in cycle counts and possibly an increase in branch mispredictions due to altered execution timing.