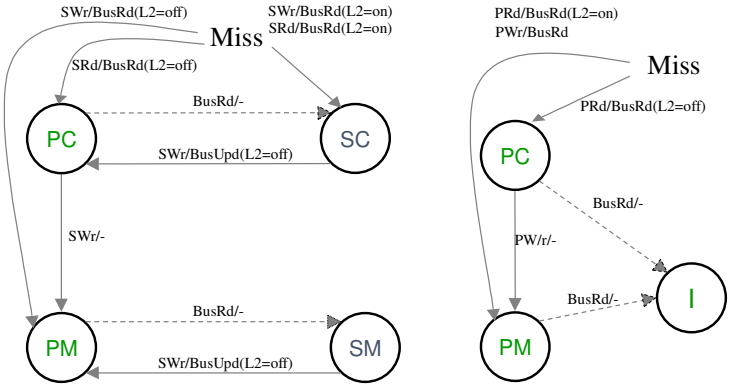
**HIGH PERFORMANCE COMPUTER ARCHITECTURE final exam 23-11-2021 MATR.NO._____**

**(reviewed from typos)** **SURNAME_____**

**FIRST NAME_____**

1) (POINTS 26/30) 1) Let's consider a shared-bus shared-memory coherence protocol (called PSCR) with the following characteristics:

- Processor Operations: Private-Read (**PRd**), Private-Write (**PWr**), Shared-Read (**SRd**) and Shared-Write (**SWr**); the private R/W are accessing private data (P-blocks) and the shared R/W are accessing shared data (S-blocks); however, if a process migrates on a different core, the shared data remains valid but the private data becomes stale and will eventually be invalidated upon a remote write that is happening on the remote core. The processor operations have a cost of 1 cycles (to bring the data into the processor from the cache).
- Bus Transactions: i) Bus-Read (**BusRd**) to read a data-block from Memory with a cost of 90 cycles; ii) Bus-Update (**BusUpd** – a single data-word goes to other caches and to memory) to propagate a PrWr on the bus with a cost of 20 cycles.
- There are 5 states: 1) the copy is not valid (**I** – Invalid); 2) the copy is in one cache only and we assume that it has been modified even if it has been just loaded, the memory is not updated (**PM** – Private Modified); 3) there is only one copy in this cache and it is consistent with the memory (**PC** – Private Clean); 4) several copies may exist and they are consistent with the memory (**SC** – Shared Clean); 5) several copies may exist, they are not consistent with the main memory and this cache (which has first written (or variant: last written) a new value) has the responsibility to provide it (**SM** – Shared Modified) to other caches.
- There are two additional bus wires: L1 always propagates the information about whether the bus transaction involves a P-block (L1=on) or an S-block (L1=off); L2 has a different meaning depending on the L1 indication: in case of a BusRd of a P-block, indicates that the copy is in modified state (so it must be loaded in PM state), but in case of a BusRd or BusUpd of an S-block it indicates if other copies still exist in the system (it acts like a "shared-line").
- Here are the complete state diagrams which describe what happens for S-blocks and P-blocks (for operations/transactions not reported, it means there is no state change).




S-blocks (L1=off) P-blocks (L1=on)

Evaluate the total cost of executing the following streams by completing the table below:

**stream-1**

| Core Operation | C1 | C2 | C3 | Bus Transaction | Data from | Cycles |
|---|---|---|---|---|---|---|
| PRd1(A) | | | | | | |
| PWr1(A) | | | | | | |
| SRd1(B) | | | | | | |
| SWr1(B) | | | | | | |
| PRd2(A) | | | | | | |
| PWr2(A) | | | | | | |
| PRd3(A) | | | | | | |
| PWr3(A) | | | | | | |
| SRd3(B) | | | | | | |
| SWr3(B) | | | | | | |
| SRd2(B) | | | | | | |
| SWr2(B) | | | | | | |
| TOTAL | | | | | | |

**stream-2**

| Core Operation | C1 | C2 | C3 | Bus Transaction | Data from | Cycles |
|---|---|---|---|---|---|---|
| PRd1(A) | | | | | | |
| PRd2(A) | | | | | | |
| PRd3(A) | | | | | | |
| SWr1(B) | | | | | | |
| SWr2(B) | | | | | | |
| SWr3(B) | | | | | | |
| SRd1(B) | | | | | | |
| SRd2(B) | | | | | | |
| PRd3(A) | | | | | | |
| PWr3(A) | | | | | | |
| PWr1(A) | | | | | | |
| TOTAL | | | | | | |

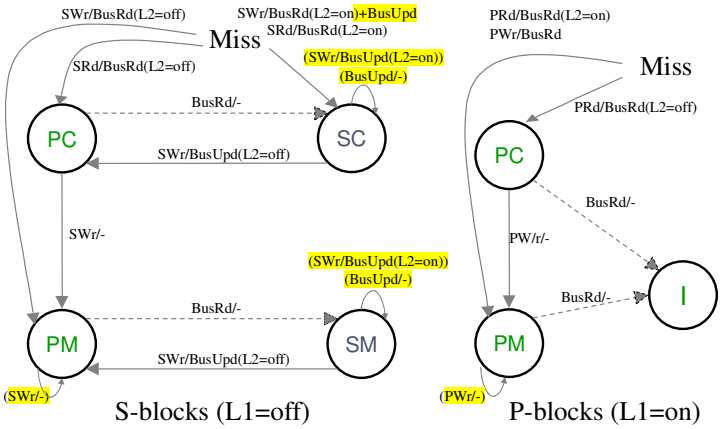
2) (POINTS 4/30) Explain with your own words and a sketch, the difference between executing a program on a CPU and executing a program on FPGAs. Also show with a sketch, the location of the binary file during the execution of the program on a CPU compared to the bitstream in the FPGA.

# HIGH PERFORMANCE COMPUTER ARCHITECTURE final exam 23-11-2021

(solution trace)

## EXERCIZE 1

PLEASE NOTE: IN YELLOW IN THE STATE DIAGRAM ARE ALSO REPORTED SOME OPEERATIONS OR TRANSACTION WICH IMPLY NOT STATE CHANGE, FOR MORE CLARITY.




S-blocks (L1=off) P-blocks (L1=on)

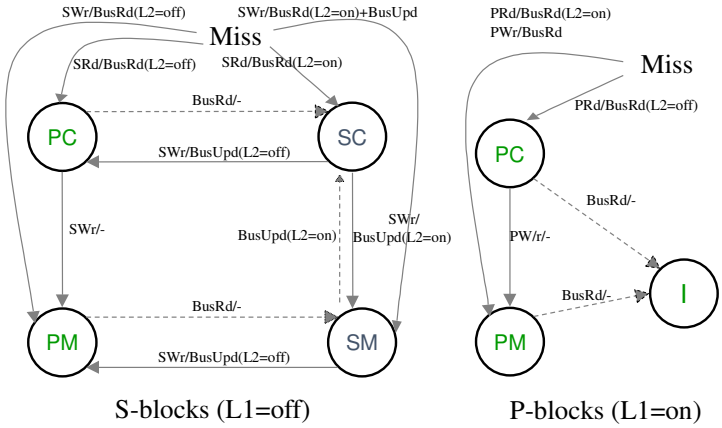
**stream-1 (A and B are two different blocks**

| Core Operation | C1 | C2 | C3 | Bus Transaction | Data from | Cycles |
|---|---|---|---|---|---|---|
| PRd1(A) | PC | I | I | BusRd(L2=off) | Mem | 90 |
| PWr1(A) | PM | I | I | - | - | 1 |
| SRd1(B) | PC | I | I | BusRd(L2=off) | Mem | 90 |
| SWr1(B) | PM | I | I | - | - | 1 |
| PRd2(A) | I | PM | I | BusRd(L2=on) | C1 | 90 |
| PWr2(A) | I | PM | I | - | - | 1 |
| PRd3(A) | | I | PM | BusRd(L2=on) | C2 | 90 |
| PWr3(A) | | I | PM | - | - | 1 |
| SRd3(B) | SM | I | SC | BusRd(L2=on) | C1 | 90 |
| SWr3(B) | SM | I | SC | BusUpd(L2=on) | - | 20 |
| SRd2(B) | SM | SC | SC | BusRd(L2=on) | C3 | 90 |
| SWr2(B) | SM | SC | SC | BusUpd(L2=on) | - | 20 |
| TOTAL | | | | | | 584 |

**stream-2 (A and B are two different blocks**

| Core Operation | C1 | C2 | C3 | Bus Transaction | Data from | Cycles |
|---|---|---|---|---|---|---|
| PRd1(A) | PC | I | I | BusRd(L2=off) | Mem | 90 |
| PRd2(A) | I | PC | I | BusRd(L2=off) | C1 | 90 |
| PRd3(A) | I | I | PC | BusRd(L2=off) | C2 | 90 |
| SWr1(B) | PM | I | I | BusRd(L2=off) | Mem | 90 |
| SWr2(B) | SM | SC | I | BusRd(L2=on)+BusUpd | C1 | 90+20 |
| SWr3(B) | SM | SC | SC | BusRd(L2=on)+BusUpd | C2 | 90+20 |
| SRd1(B) | SM | SC | SC | - | - | 1 |
| SRd2(B) | SM | SC | SC | - | - | 1 |
| PRd3(A) | I | I | PC | - | - | 1 |
| PWr3(A) | I | I | PM | - | - | 1 |
| PWr1(A) | PM | I | I | BusRd(L2=on) | C3 | 90 |
| TOTAL | | | | | | 674 |

VARIANT




S-blocks (L1=off) P-blocks (L1=on)

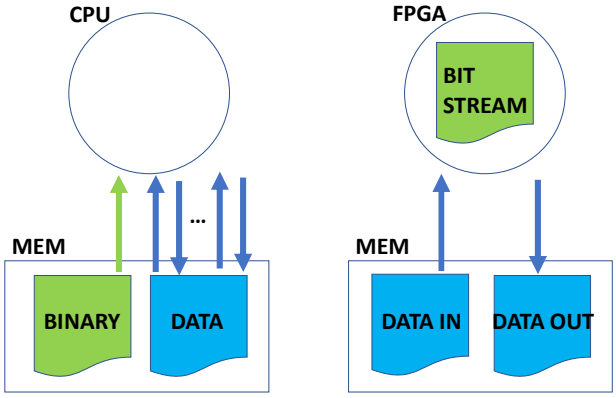
**stream-1 (A and B are two different blocks**

| Core Operation | C1 | C2 | C3 | Bus Transaction | Data from | Cycles |
|---|---|---|---|---|---|---|
| PRd1(A) | PC | I | I | BusRd(L2=off) | Mem | 90 |
| PWr1(A) | PM | I | I | - | - | 1 |
| SRd1(B) | PC | I | I | BusRd(L2=off) | Mem | 90 |
| SWr1(B) | PM | I | I | - | - | 1 |
| PRd2(A) | I | PM | I | BusRd(L2=on) | C1 | 90 |
| PWr2(A) | I | PM | I | - | - | 1 |
| PRd3(A) | | I | PM | BusRd(L2=on) | C2 | 90 |
| PWr3(A) | | I | PM | - | - | 1 |
| SRd3(B) | SM | I | SC | BusRd(L2=on) | C1 | 90 |
| SWr3(B) | SC | I | SM | BusUpd(L2=on) | - | 20 |
| SRd2(B) | SC | SC | SM | BusRd(L2=on) | C3 | 90 |
| SWr2(B) | SC | SM | SC | BusUpd(L2=on) | - | 20 |
| TOTAL | | | | | | 584 |

**stream-2 (A and B are two different blocks**

| Core Operation | C1 | C2 | C3 | Bus Transaction | Data from | Cycles |
|---|---|---|---|---|---|---|
| PRd1(A) | PC | I | I | BusRd(L2=off) | Mem | 90 |
| PRd2(A) | I | PC | I | BusRd(L2=off) | C1 | 90 |
| PRd3(A) | I | I | PC | BusRd(L2=off) | C2 | 90 |
| SWr1(B) | PM | I | I | BusRd(L2=off) | Mem | 90 |
| SWr2(B) | SC | SM | I | BusRd(L2=on)+BusUpd | C1 | 90+20 |
| SWr3(B) | SC | SC | SM | BusRd(L2=on)+BusUpd | C2 | 90+20 |
| SRd1(B) | SC | SC | SM | - | - | 1 |
| SRd2(B) | SC | SC | SM | - | - | 1 |
| PRd3(A) | I | I | PC | - | - | 1 |
| PWr3(A) | I | I | PM | - | - | 1 |
| PWr1(A) | PM | I | I | BusRd(L2=on) | C3 | 90 |
| TOTAL | | | | | | 674 |

## EXERCIZE 2

Running a program on a CPU involves executing one instruction after the other in a temporal fashion. The execution on an FPGA involves streaming data into a fixed function and streaming data out of it: the execution happens in a spatial fashion.

The position of the binary in the CPU is the memory, whilst in the FPGA the program is the function that has been stored in the FPGA itself. There is an important difference also on the data: in the CPU case, data is read and written several times from and to memory during time, whilst on the FPGA the data is typically coming from a predefined source once and is sent to a predefined destination once.