Title of the Project: Lab6 Multi-Core

Introduction: In this lab, we modify CPU to multi-core CPU from lab5.

Question:

Q1: How do you separate the program for the two cores?

A: One core count the left column of answer_matrix(matrix1 * the left column of matrix2),and the other core count the right column of answer_matrix(matrix1 * the right column of matrix2).

Q2: Assumed that programmers do not know the platform architecture (i.e. single core or multi-core) how can programmers manage their program partition?

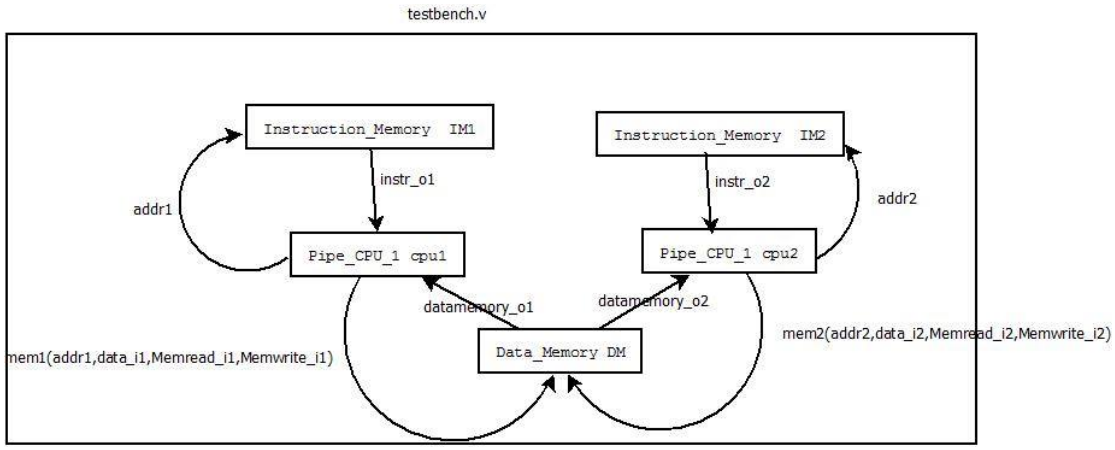
A: We should divide the program into many threads.

Q3: Assumed that each core has private cache. If core1 write a new data at address 0x123, how could core 2 get the new data from 0x123? (hint: coherence)

A: If the private cache of each core has “coference” then core2 can get the new data from 0x123.

Q4: Draw the detail architecture and describe your design in your report.

A:

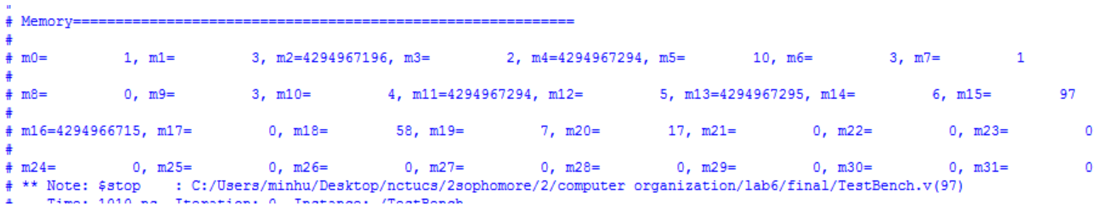


Procedure:

We do two Instruction_Memory and Data_Memoryin TestBench, so we don’t have the” Instruction_Memory” blocks,” Data_Memory” block in “Pipe_CPU”. As this condition,we need to input the instruction and the data from “Data_Memory” in “Pipe_CPU”.

Two core use a “Data_Memory” together, so it should have two same type of input and output port(as the ports in single-core) for two core.

Result:



```
"
# Memory============================================================
#
# m0=         1, m1=         3, m2=4294967196, m3=         2, m4=4294967294, m5=        10, m6=         3, m7=         1
#
# m8=         0, m9=         3, m10=         4, m11=4294967294, m12=         5, m13=4294967295, m14=         6, m15=        97
#
# m16=4294966715, m17=         0, m18=        58, m19=         7, m20=        17, m21=         0, m22=         0, m23=         0
#
# m24=         0, m25=         0, m26=         0, m27=         0, m28=         0, m29=         0, m30=         0, m31=         0
# ** Note: $stop    : C:/Users/minhu/Desktop/nctucs/2sophomore/2/computer organization/lab6/final/TestBench.v(97)
#    Time: 1010 ns  Iteration: 0  Instance: /TestBench
```

Conclusion:

In order to avoid error, we should connect each blocks carefully.