Simulation Report

MIPS CPU

**Appendix C**

|  |  |
| --- | --- |
| Students: | Otabek Sherman -326868775 |
|  | Lidor Cohen - 305782732 |

Answer the following questions.

C.1) What are the limitations due to the pipeline latency of the following combinations (assume Data Forwarding already exists):

* beq after add where the add Rd is the beq Rt
* beq after lw where the lw Rt is the beq Rs

Use a similar figure to Fig.2 and Fig. 3 to demonstrate your answers. Explain your answers!

C.1.a - beq after add where the add Rd is the beq Rt

WB

EX

ID

IF

WB

EX

ID

IF

WB

EX

ID

IF

add **$3**,$5,$8

WB

EX

ID

IF

WB

EX

ID

IF

CK

MEM

MEM

MEM

MEM

MEM

EX



In case of add instruction before beq instruction and Data Forwarding exist, so we will wait 2 clocks between the beq and add instructions to get the correct result of the add instruction. Beq instruction can access the data in the ID step, because the data is already written to GPR in WB step of the add instruction.

C.1.b - beq after lw where the lw Rt is the beq Rs

WB

EX

ID

IF

WB

EX

ID

IF

WB

EX

ID

IF

lw **$3**,16($10)

WB

EX

ID

IF

WB

EX

ID

IF

CK

MEM

MEM

MEM

MEM

MEM

EX



In case of lw instruction is before beq and Data Forwarding exists, we need to wait 2 nop to get the correct data. In lw instruction the data is stored from DMEM to the address stored in the register $3. In the beq instruction in the step ID we can access the data.

C.2) What are the limitations of all cases of C.1 after you add the Branch Forwarding? . Explain your answers!

C.2.a - beq after add where the add Rd is the beq Rt

WB

EX

ID

IF

WB

EX

ID

IF

WB

EX

ID

IF

add **$3**,$5,$8

WB

EX

ID

IF

WB

EX

ID

IF

CK

MEM

MEM

MEM

MEM

MEM

EX



Add instruction comes before beq. We will wait the 1 clock until the result of add instruction is available. In the step MEM of the add instruction the result is available and we can load it to the rt value of the beq instruction in the ID step.

C.2.b - beq after lw where the lw Rt is the beq Rs

WB

EX

ID

IF

WB

EX

ID

IF

WB

EX

ID

IF

lw **$3**,16($10)

WB

EX

ID

IF

WB

EX

ID

IF

CK

MEM

MEM

MEM

MEM

MEM

EX



Lw instruction comes before beq instruction, so we must to wait when data is available from DMEM, then we can utilize it by the beq instruction.

We will wait 2 clocks until the lw instruction will access the data.

C.3) Why can’t we check the result of the previous instruction (time slot n-1) by a beq instruction following it (time slot n)?

Because the previous instruction will reach the EX step and will not finish the calculation in the ALU unit.

To check the result of the previous instruction we need to wait 1 additional clock.

C.4) List all of the limitations for Assembly programmer you can think of that still exist after adding the Data & Branch Forwarding circuits. . Explain your answer!

* In all R-Type and I-Type instructions we need to wait at least 1 clock cycle to get the correct result from the ALU.
* To load 32 bits we need to use 2 instructions ORI and LUI.

C5) What is the shortest loop code possible (not an infinite loop)? Any limitations? Explain in detail

We can implement the shortest loop code by j instruction. The program will fetch from the PC register the j instruction that will jump back to the PC register.