

Computation Structures 3: Computer Organization

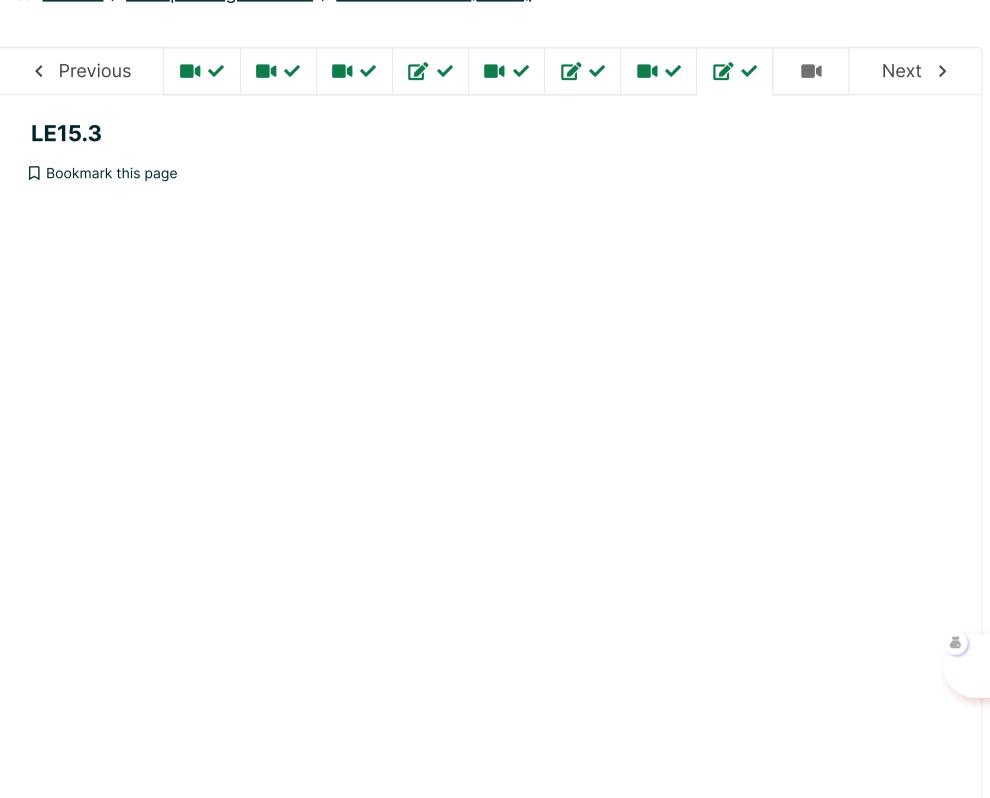
<u>Help</u>





<u>Course</u> <u>Progress</u> <u>Dates</u> <u>Discussion</u>

☆ Course / 15. Pipelining the Beta / Lecture Videos (50:06)



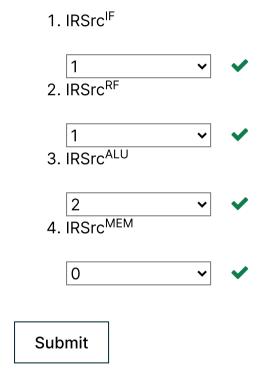
For all Beta related questions, you should make use of the <u>Beta documentation</u>, the <u>Beta Instruction</u> <u>Summary</u>, the <u>Unpipelined Beta Diagram</u> and the <u>Pipelined Beta Diagram</u>.

LE15.3.1: Exceptions

1.0/1.0 point (ungraded)

An ambitious 6.004x student has enhanced the 5-stage pipelined Beta to trigger an exception in the ALU stage when there's an attempt to divide by zero, i.e., when the B operand of the ALU is zero and the opcode of the instruction in the ALU stage is DIV or DIVC.

For each of the following control signals, indicate its value during the clock cycle when the divide-by-zero exception occurs.



LE15.3.2: Bypassing, stalls, annulment

1/1 point (ungraded)

The loop below has been executing for a while on our standard 5-stage pipelined Beta with full bypassing and speculation on branches (i.e., instruction in IF stage is annulled when there's a taken branch in the RF stage). The pipeline diagram below shows the opcode of the instruction in each pipeline stage during 9 consecutive cycles of execution.

```
LOOP: LD(R0, 0, R2)
SHLC(R1, 1, R1)
ADD(R1, R2, R1)
ADDC(R0, 4, R0)
CMPEQC(R0, aend, R2)
BNE(R2, LOOP)
ST(R1, hash, R31)
...
```

Cycle#	200	201	202	203	204	205	206	207	208
IF	LD	SHLC	ADD	ADDC	ADDC	CMPEQC	BNE	ST	LD
RF	NOP	LD	SHLC	ADD	ADD	ADDC	CMPEQC	BNE	NOP
ALU	BNE	NOP	LD	SHLC	NOP	ADD	ADDC	CMPEQC	BNE
MEM	CMPEQC	BNE	NOP	LD	SHLC	NOP	ADD	ADDC	CMPEQC
WB	ADDC	CMPEQC	BNE	NOP	LD	SHLC	NOP	ADD	ADDC

1. For each cycle, specify whether or not a bypass path was used to get the data from a later pipeline stage to the RF stage. If multiple bypass paths are used in a particular cycle, then select all that apply.

Cycle 200:

✓	No bypass paths used	
	Bypass from ALU to RF	
	Bypass from MEM to RF	
	Bypass from WB to RF	
~		
Cycle	201:	
~	No bypass paths used	
	Bypass from ALU to RF	
	Bypass from MEM to RF	
	Bypass from WB to RF	
~		
Cycle	202:	
✓	No bypass paths used	
	Bypass from ALU to RF	
	Bypass from MEM to RF	
	Bypass from WB to RF	•
~		
Cycle	203:	
✓	No bypass paths used	
	Bypass from ALU to RF	
	Bypass from MEM to RF	
	Bypass from WB to RF	
~		
Cycle	204:	
	No bypass paths used	
	Bypass from ALU to RF	
✓	Bypass from MEM to RF	
✓	Bypass from WB to RF	
. 🙍		

	Lecture Videos (50:06) 15. Pipelining the Beta Computation Structures 3: Computer Organization edX
Cycle	205:
✓	No bypass paths used
	Bypass from ALU to RF
	Bypass from MEM to RF
	Bypass from WB to RF
~	
Cycle	206:
	No bypass paths used
✓	Bypass from ALU to RF
	Bypass from MEM to RF
	Bypass from WB to RF
~	
Cycle	207:
	No bypass paths used
✓	Bypass from ALU to RF
	Bypass from MEM to RF
	Bypass from WB to RF
~	
Cycle	208:
✓	No bypass paths used
	Bypass from ALU to RF
	Bypass from MEM to RF
	Bypass from WB to RF
	e following questions think carefully about when a signal would be asserted in order to produce the you see in the pipeline diagram. Select all the cycles that apply, or select NONE if it never occurs.
During	g which cycle(s), if any, would the IRSrc ^{IF} signal be 1?
	200
	201
	202

204

205

206

208

NONE

207

3.	During	g which cycle(s), if any, would the IRSrc ^{RF} signal be 1?
		200
		201
		202
	~	203
		204
		205
		206
		207
		208
		NONE
	~	
4.	During	g which cycle(s), if any, would the IRSrc ^{ALU} signal be 1?
		200
		201
		202
		203
		204
		205
		206
		207
		208
	~	NONE
cour	se-v1:MIT	$x+6.004.3x_2+1T2017/block-v1:MITx+6.004.3x_2+1T2017+type@sequential+block@c3s1/block-v1:MITx+6.004.3x_2+1T2017+type@vertical+block@c3s1v8$

The ADDC then proceeds since it does not have any data hazards. The CMPE, so in cycle 206 it uses the bypass path from the ALU stage to get the result of to use a bypass path from the ALU stage to read the value of R2 that is production. Finally, in cycle 207, the ST which was fetched following the branch must his taken. This is set to 1 when the instruction in the IF stage has to be annulled because of a fine in cycle 207. The is set to 1 when STALL is set to 1. STALL is set to 1 when a register value needing is not yet available in the data path or register file. This happens in cycle 200 is set to 1 when the instruction in the ALU has to be annulled because of an analysis, i.e., the MEM stage. That doesn't happen in this instruction sequence. The importance of the instruction is the problem with the problem of the Beta / LE15.3	f the ADDC. The BNE also ced by the CMPEQC be annulled because the taken branch. This ded by the instruction in the 13.
, so in cycle 206 it uses the bypass path from the ALU stage to get the result of to use a bypass path from the ALU stage to read the value of R2 that is production. Finally, in cycle 207, the ST which was fetched following the branch must in is taken. First is set to 1 when the instruction in the IF stage has to be annulled because of a case in cycle 207. First is set to 1 when STALL is set to 1. STALL is set to 1 when a register value needing is not yet available in the data path or register file. This happens in cycle 20 is set to 1 when the instruction in the ALU has to be annulled because of an instruction in the ALU has to be annulled because of an instruction in the MEM stage. That doesn't happen in this instruction sequence.	f the ADDC. The BNE also ced by the CMPEQC be annulled because the taken branch. This ded by the instruction in the 13.
to use a bypass path from the ALU stage to get the result of to use a bypass path from the ALU stage to read the value of R2 that is production. Finally, in cycle 207, the ST which was fetched following the branch must be is taken. This is set to 1 when the instruction in the IF stage has to be annulled because of a sens in cycle 207. This is set to 1 when STALL is set to 1. STALL is set to 1 when a register value needing is not yet available in the data path or register file. This happens in cycle 20 and is set to 1 when the instruction in the ALU has to be annulled because of an is, i.e., the MEM stage. That doesn't happen in this instruction sequence.	f the ADDC. The BNE also ced by the CMPEQC be annulled because the taken branch. This ded by the instruction in the 13.
, so in cycle 206 it uses the bypass path from the ALU stage to get the result of to use a bypass path from the ALU stage to read the value of R2 that is production. Finally, in cycle 207, the ST which was fetched following the branch must have is taken. This is set to 1 when the instruction in the IF stage has to be annulled because of a constant in cycle 207. This is set to 1 when STALL is set to 1. STALL is set to 1 when a register value needing is not yet available in the data path or register file. This happens in cycle 200 is set to 1 when the instruction in the ALU has to be annulled because of an annulled because of annulled becaus	f the ADDC. The BNE also ced by the CMPEQC be annulled because the taken branch. This ded by the instruction in the 13.
ata hazards. The ADD needs to read R1 and R2 so it depends on the results of t tion. The result from the LD cannot be bypassed until the LD is in the WB stage neans that in cycle 203, we need to stall the ADD because it cannot yet read its he result of the LD is bypassed from the WB stage and the result of the SHLC i	which occurs in cycle 204. soperands. Then in cycle s bypassed from the MEM
nation D is fetched in cycle 200 and proceeds down the pipeline stages. The SHLC reserves have a specific depends on the results of the second stages.	
NONE	
208	
207	
206	
204	
203	
202	
201	

Next >

Previous



edX

About

Affiliates

edX for Business

Open edX

<u>Careers</u>

<u>News</u>

Legal

Terms of Service & Honor Code

Privacy Policy

<u>Accessibility Policy</u>

Trademark Policy

<u>Sitemap</u>

Cookie Policy

Your Privacy Choices

Connect

<u>Idea Hub</u>

Contact Us

Help Center

<u>Security</u>

Media Kit

















© 2024 edX LLC. All rights reserved.

深圳市恒宇博科技有限公司 <u>粤ICP备17044299号-2</u>