Outline

- Introduction
- Architecture of a RISC Processor
- Implementation

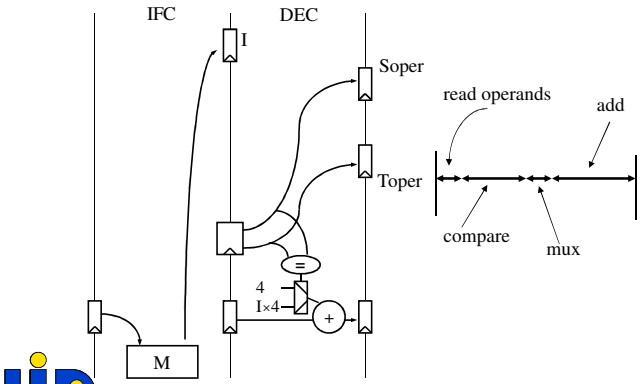
Implementation

Implementation of DEC cycle

- Next instruction address calculation

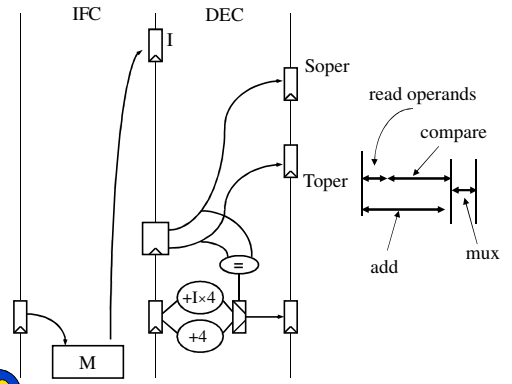
Implementation

Example : Beq rs, rt, label



Implementation

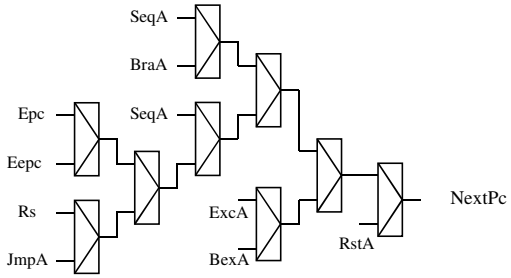
Example : Beq rs, rt, label



Next instruction address

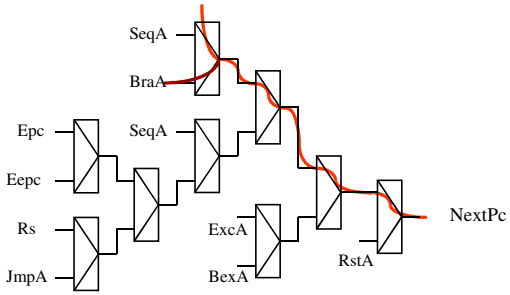
- Sequential address (SeqA)
- Branch target address (BraA)
- Jump address (JmpA)
- Source register (Rs)
- `0xBFC0 0000` (RstA)
- `0xBFC0 0380` (BexA)
- Exception base register (ExcA)
- Return address (Epc)
- Return address (Eepc)

Next instruction address



Reset > Exception > Instruction

Next instruction address



Propagation delays on commands
Propagation delays on data

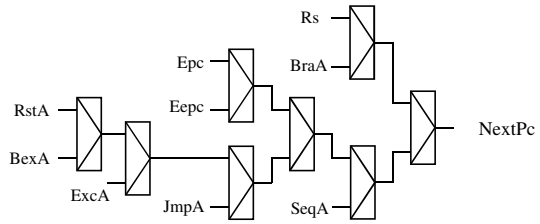
Next instruction address

Considering data propagation delays

BraA > SeqA, Rs >> Epc, Eepc, JmpA, ExcA > RstA, BexA

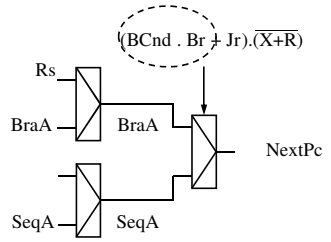
Next instruction address

Considering data propagation delays



Next instruction address

Considering data propagation delays



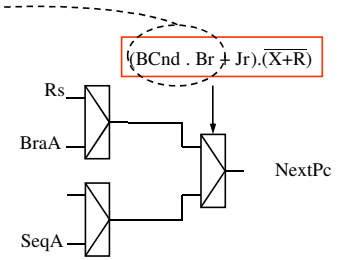
Next instruction address

- Sequential address (SeqA)
- Branch target address (BraA)

	Beq	Bne	Bltz	Bgez	Blez	Bgtz
Condition	Rs = Rt	Rs ≠ Rt	Rs < 0	Rs ≥ 0	Rs ≤ 0	Rs > 0
True	BraA	BraA	BraA	BraA	BraA	BraA
False	SeqA	SeqA	SeqA	SeqA	SeqA	SeqA

Next instruction address

$Rs = Rt$ if Beq
$Rs \neq Rt$ if Bne
$Rs < 0$ if Bltz
$Rs \geq 0$ if Bgez
$Rs \leq 0$ if Blez
$Rs > 0$ if Bgtz



Propagation delays on commands

Next instruction address

- Sequential address (SeqA)
- Branch target address (BraA)

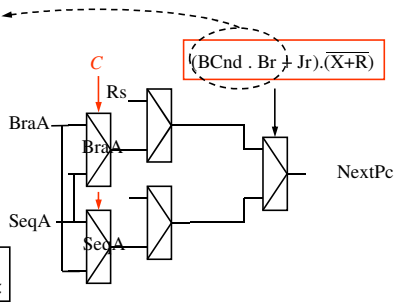
	Beq	Bne	Bltz	Bgez	Blez	Bgtz
Condition	Rs = Rt	Rs = Rt	Rs < 0	Rs < 0	Rs ≤ 0	Rs ≤ 0
True	BraA	SeqA	BraA	SeqA	BraA	SeqA
False	SeqA	BraA	SeqA	BraA	SeqA	BraA

3 conditions instead of 6 - swapped data

Next instruction address

Rs = Rt *if* ***Beq***, *Bne*
Rs ≠ Rt *if* *Bne*
Rs < 0 *if* ***Bltz***, *Bgez*
Rs ≥ 0 *if* *Bgez*
Rs ≤ 0 *if* ***Blez***, *Bgtz*
Rs > 0 *if* *Bgtz*

C = 1 *if* *Beq, Bltz, Blez*
C = 0 *if* *Bne, Bgez, Bgtz*



Propagation delays on commands

Next instruction address

- Sequential address (SeqA)
- Branch target address (BraA)

	Beq	Bne	Bltz	Bgez	Blez	Bgtz
Condition	Rs = Rt	Rs = Rt	Rs < 0	Rs < 0	Rs ≤ 0	Rs ≤ 0
True	BraA	SeqA	BraA	SeqA	BraA	SeqA
False	SeqA	BraA	SeqA	BraA	SeqA	BraA

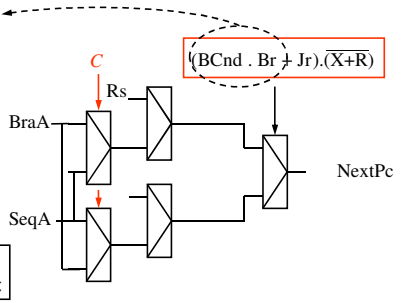
Rs < 0
Rs = 0

3 conditions instead of 6 - swapped data

Next instruction address

Rs = Rt *if* ***Beq, Bne,***
Blez, Bgtz
Rs < 0 *if* ***Bltz, Bgez,***
Blez, Bgtz
Rs ≤ 0 *if* *Blez, Bgtz*

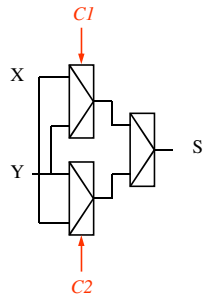
C = 1 *if* *Beq, Bltz, Blez*
C = 0 *if* *Bne, Bgez, Bgtz*



Propagation delays on commands

Next instruction address

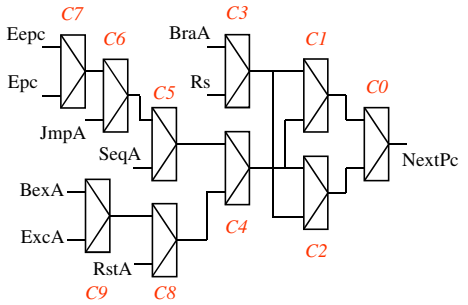
if C1 = 1 and C2 = 0 ⇒ S = X
if C1 = 0 and C2 = 1 ⇒ S = Y



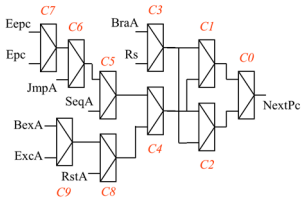
Next instruction address

Putting all together

C7 = Status (2)
C6 = Eret
C9 = Status (22)
C8 = $\overline{Reset}$
C5 = Eret + J



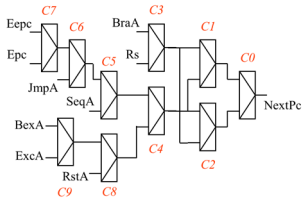
Next instruction address



	C0		C1	C2		C3	C4
Beq T	1		1	1		1	1
Beq F	0		1	1		1	1
Bne T	0		0	0		1	1
Bne F	1		0	0		1	1
Bltz T	-		1	0		1	1
Bltz F	-		0	1		1	1
Blez T	1	if Rs=Rt	1	0	if Rs<0	1	1
Blez F	0		0	1		1	1
Bgtz T	0		0	1		1	1
Bgtz F	1		1	0		1	1
Bgez T	-		0	1		1	1
Bgez F	-		1	0		1	1
Jr	-		1	0		0	-
J	-		0	1		-	1
Eret	-		0	1		-	1
Seq	-		0	1		-	1
XR	-		0	1		-	0

In Blez, Bgtz, the Rt field is 0 (Rt = R0)

Next instruction address



C3 = Br
C4 = $\overline{XR}$
C2 = (Bltz+Bgez+ Blez+Bgtz) .(Rs<0) + Beq + J + Eret + Seq + XR
C1 = (Bltz+Bgez+ Blez+Bgtz) .($\overline{Rs<0}$) + Beq + Jr
C0 = (Rs=Rt)

	C0		C1	C2		C3	C4
Beq T	1		1	1		1	1
Beq F	0		1	1		1	1
Bne T	0		0	0		1	1
Bne F	1		0	0		1	1
Bltz T	-		1	0		1	1
Bltz F	-		0	1		1	1
Blez T	1	if Rs=Rt	1	0	if Rs<0	1	1
Blez F	0		0	1		1	1
Bgtz T	0		0	1		1	1
Bgtz F	1		1	0		1	1
Bgez T	-		0	1		1	1
Bgez F	-		1	0		1	1
Jr	-		1	0		0	-
J	-		0	1		-	1
Eret	-		0	1		-	1
Seq	-		0	1		-	1
XR	-		0	1		-	0