Computer Architectures
Exam of April 28, 2020
Part 1

Exe #1, Point #1

Let consider the Branch Prediction Mechanism based on the Branch Target Buffer (BTB).

You are requested to

1. *Describe the architecture of a BTB*

The BTB is a table composed of n entries, each composed of 2 m-bit fields: Address and Target.

An additional 1 bit field may be present.

Exe #1, Point #2

2. *Describe in details the behavior of a BTB: when it is accessed, which input and output information are involved with each access*

The BTB is accessed each time an instruction is fetched.

Using the least significant log n bits of the instruction address (excluding the word alignment bits) an entry is selected. The Address field of the entry is compared with the address of the fetched instruction: if they match, the Target field is uploaded in the PC.

When the instruction is completed, the BTB is possibly updated.

Exe #1, Point #3

3. *Assuming that the processor uses 32 bit addresses, each instruction is 4 byte wide, and the BTB is composed of 8 entries, clarify the content and size of the fields composing each BTB entry*

The BTB in this case is composed of 8 entries, each storing 32 bits (Address) + 32 bits (Target).

Since instructions are aligned to words (4 bytes), the 2 least significant bits of each field can be omitted (being 0).

Exe #1, Point #4: BTB initial content

address	target
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000

Exe #1, Point #4: BTB content after l.d execution

l.d address	
0x00A50050	0000 0000 1010 0101 0000 0000 010**1 00**00

Entry #4 is accessed

address	target
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000

Exe #1, Point #4: BTB content after bnez execution

bnez address	
0x00A50054	0000 0000 1010 0101 0000 0000 010**1 01**00

address	target
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x00A50054	0x00A60050
0x0000000	0x0000000
0x0000000	0x0000000

Entry #5 is accessed

Exe #1, Point #4: BTB content after bez execution

bez address	
0x00A60050	0000 0000 1010 0110 0000 0000 010**1** **00**00

Entry #4 is accessed

address	**target**
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x00A50054	0x00A60050
0x0000000	0x0000000
0x0000000	0x0000000

Exe #1, Point #4: BTB final content

address	target
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x0000000	0x0000000
0x00A50054	0x00A60050
0x0000000	0x0000000
0x0000000	0x0000000

Exe #2

# iteration		ISSUE	EXE	MEM	CDB	COMMIT	Notes
1	l.d f1,v1(r1)	1	2m	3	4	5	
1	l.d f2,v2(r1)	1	3m	4	5	6	
1	l.d f3,v3(r1)	2	4m	5	6	7	
1	div.d f5, f3, f11	2	7d		15	16	Wait for f3
1	sub.d f4, f1, f2	3	6a		8	17	Wait for f2
1	add.d f6, f4, f5	3	16a		18	19	Wait for f5
1	div.d f7,f6,f12	4	23d		31	32	Wait for f6, then for div unit
1	s.d f7,v4(r1)	4	5m			33	
1	daddui r1,r1,8	5	6i		7	34	
1	daddi r2,r2,-1	5	7i		9	35	
1	bnez r2,loop	6	10j			36	Wait for r2
2	l.d f1,v1(r1)	7	8m	9	10	37	
2	l.d f2,v2(r1)	7	9m	10	11	38	
2	l.d f3,v3(r1)	8	10m	11	12	39	
2	div.d f5, f3, f11	8	15d		23	40	Wait for f3, then for div unit
2	sub.d f4, f1, f2	9	12a		14	41	Wait for f2
2	add.d f6, f4, f5	9	24a		26	42	Wait for f5
2	div.d f7,f6,f12	10	31d		39	43	Wait for f6, then for div unit
2	s.d f7,v4(r1)	10	11m			44	
2	daddui r1,r1,8	11	12i		13	45	
2	daddi r2,r2,-1	11	13i		16	46	
2	bnez r2,loop	12	17j			47	Wait for r2