

# Computer Architectures

## Exam of 17.5.2023 - part I

First name, Last name, ID.....

### Question #2

Let's consider a MIPS architecture using a *Branch History Table* (BHT) composed of 16 1-bit entries. Let's assume that this architecture executes the following code, which counts the number of values equal to zero and one contained in vector *vec* and then writes the result in the variables *res0* and *res1*. For every instruction, the hexadecimal address of the memory cell storing the instruction is reported. Assuming that when the execution of the code fragment the BHT is full of null values (corresponding to the prediction Not Taken) you are asked to compute:

- The number of mispredicted branches during the execution of the code.
- The BHT content when the execution finishes (using the table reported on the next page).

For all computations, it is suggested the usage of the table on the next page. Write in the highlighted cells whether the result of the prediction of the current branch and the real behavior (result) of the software is *Taken* (T) or *Not Taken* (NT).

*Hint: To calculate the BHT entry corresponding to each branch instruction, remember that you should exclude the last two bits from the instruction address as they are always equal to 0.*

vec:	. byte	0, 2, 2, 1, 0, 0, 1	# input vector
res0:	.space	1	# number of values equal to zero
res1:	.space	1	# number of values equal to one
...			
0x0000		daddui r1, r0, 1	# initialize the value used as a comparator
0x0004		daddui r2, r0, 0	# initialize the pointer
0x0008		daddui r3, r0, 0	# initialize the counter of elements equal to zero
0x000c		daddui r4, r0, 0	# initialize the counter of elements equal to one
0x0010		daddui r5, r0, 7	# initialize the value containing the size of the vector
0x0014	cyc:	beq r2, r5, term	# condition for exiting the cycle
0x0018		lb r6, vec(r2)	# load an element from vec
0x001c		beq r6, r0, eq0	# check if the loaded value is equal to zero
0x0020		beq r6, r1, eq1	# check if the loaded value is equal to one
0x0024		j nxt	# values different from zero or one are skipped
0x0028	eq0:	daddui r3, r3, 1	# increment the counter of elements equal to zero
0x002c		j nxt	# go to the next cycle
0x0030	eq1:	daddui r4, r4, 1	# increment the counter of elements equal to one
0x0034	nxt:	daddui r2, r2, 1	# increment the pointer
0x0038		j cyc	# next cycle
0x003c	term:	sb r3, res0(r2)	# store the result
0x0040		sb r4, res1(r2)	# store the result
0x0044	halt		# termination of the program

# Computer Architectures

## Exam of 17.5.2023 - part I

First name, Last name, ID.....

Address	Code	BHT	Iteration #1		Iteration #2		Iteration #3		Iteration #4	
		entry #	prediction	result	prediction	result	prediction	result	prediction	result
0x0000	daddui r1, r0, 1									
0x0004	daddui r2, r0, 0									
0x0008	daddui r3, r0, 0									
0x000c	daddui r4, r0, 0									
0x0010	daddui r5, r0, 6									
0x0014	cyc: beq r2, r5, term	5	NT	NT	NT	NT	NT	NT	NT	NT
0x0018	lb r6, vec(r2)									
0x001c	beq r6, r0, eq0	7	NT	T	T	NT	NT	NT	NT	NT
0x0020	beq r6, r1, eq1	8			NT	NT	NT	NT	NT	T
0x0024	j nxt									
0x0028	eq0: daddui r3, r3, 1									
0x002c	j nxt									
0x0030	eq1: daddui r4, r4, 1									
0x0034	nxt: daddui r2, r2, 1									
0x0038	j cyc									
0x003c	term: sb r3, res0(r2)									
0x0040	sb r4, res1(r2)									
0x0044	halt									

Address	Code	BHT	Iteration #5		Iteration #6		Iteration #7		Iteration #8	
		entry #	prediction	result	prediction	result	prediction	result	prediction	result
0x0000	daddui r1, r0, 1									
0x0004	daddui r2, r0, 0									
0x0008	daddui r3, r0, 0									
0x000c	daddui r4, r0, 0									
0x0010	daddui r5, r0, 6									
0x0014	cyc: beq r2, r5, term	5	NT	NT	NT	NT	NT	NT	NT	T
0x0018	lb r6, vec(r2)									
0x001c	beq r6, r0, eq0	7	NT	T	T	T	T	NT		
0x0020	beq r6, r1, eq1	8					T	T		
0x0024	j nxt									
0x0028	eq0: daddui r3, r3, 1									
0x002c	j nxt									
0x0030	eq1: daddui r4, r4, 1									
0x0034	nxt: daddui r2, r2, 1									
0x0038	j cyc									
0x003c	term: sb r3, res0(r2)									
0x0040	sb r4, res1(r2)									
0x0044	halt									

The number of mispredicted branches during the execution of the code is: 6

### BHT - Final content

Entry 0	0	Entry 4	0	Entry 8	1	Entry 12	0
Entry 1	0	Entry 5	1	Entry 9	0	Entry 13	0
Entry 2	0	Entry 6	0	Entry 10	0	Entry 14	0
Entry 3	0	Entry 7	0	Entry 11	0	Entry 15	0