

BPU Examples

E. Sanchez

Politecnico di Torino
Dipartimento di Automatica e Informatica

BHT Example

Considering a 2-bit saturating counter BHT of 1K entries; and assuming that the processor executes the following code fragment, determine the BHT final state and calculate the misprediction ratio for the presented case.

```
for (i=1; i<=100; i++) {  
    read_data();  
    if (aa==2)  
        aa = 0;  
    if (bb==2)  
        bb = 0;  
    if (aa == bb)  
    {...}  
}
```

BHT Example

Considering a 2-bit saturating counter BHT of 1K entries; and assuming that the processor executes the following code fragment, determine the BHT final state and calculate the misprediction ratio for the presented case.

```
for(i=1;i<=100;i++){
    read_data();
    if (aa==2)
        aa = 0;
    if (bb==2)
        bb = 0;
    if (aa == bb)
        {...}
}
```

```

L0:  ...
      addi    x3, x1, -2
      bnez    x3, L1
      add     x1, x0, x0
L1:  addi    x3, x2, -2
      bnez    x3, L2
      add     x2, x0, x0
L2:  sub     x3, x1, x2
      bnez    x3, L3
      ...
L3:  ...
      addi    x4, x4, -1
      bnez    x4, L0
      ...
```

BHT Example

Considering a 2-bit saturating counter BHT of 1K entries; and assuming that the processor executes the following code fragment, determine the BHT final state and calculate the misprediction ratio for the presented case.

Case 1:

-Program inputs for aa and bb are always different than 2.

Iteration 1

aa = 4
bb = 3

address	Instruction		BHT	Prediction	misP. counter
0x0000	L0:	...	0	NT	
...		...	0	NT	
0x0010		addi x3, x1, -2	0	NT	
0x0014		bnez x3, L1	0	NT	
0x0018		add x1, x0, x0	0	NT	
0x001C	L1:	addi x3, x2, -2	0	NT	
0x0020		bnez x3, L2	0	NT	
0x0024		add x2, x0, x0	0	NT	
0x0028	L2:	sub x3, x1, x2	0	NT	
0x002C		bnez x3, L3	0	NT	
0x0030		...	0	NT	
0x0034	L3:	...	0	NT	
0x0038		addi x4, x4, -1	0	NT	
0x003C		bnez x4, L0	0	NT	
0x0040		...	0	NT	

Iteration 1

aa = 4
bb = 3

address	Instruction		BHT	Prediction	misP. counter
0x0000	L0:	...	0	NT	
...		...	0	NT	
0x0010		addi x3, x1, -2	0	NT	
0x0014		bnez x3, L1	1	NT	1
0x0018		add x1, x0, x0	0	NT	
0x001C	L1:	addi x3, x2, -2	0	NT	
0x0020		bnez x3, L2	0	NT	
0x0024		add x2, x0, x0	0	NT	
0x0028	L2:	sub x3, x1, x2	0	NT	
0x002C		bnez x3, L3	0	NT	
0x0030		...	0	NT	
0x0034	L3:	...	0	NT	
0x0038		addi x4, x4, -1	0	NT	
0x003C		bnez x4, L0	0	NT	
0x0040		...	0	NT	

Iteration 1

aa = 4
bb = 3

address	Instruction		BHT	Prediction	misP. counter
0x0000	L0:	...	0	NT	
...		...	0	NT	
0x0010		addi x3, x1, -2	0	NT	
0x0014		bnez x3, L1	1	NT	1
0x0018		add x1, x0, x0	0	NT	
0x001C	L1:	addi x3, x2, -2	0	NT	
0x0020		bnez x3, L2	1	NT	1
0x0024		add x2, x0, x0	0	NT	
0x0028	L2:	sub x3, x1, x2	0	NT	
0x002C		bnez x3, L3	0	NT	
0x0030		...	0	NT	
0x0034	L3:	...	0	NT	
0x0038		addi x4, x4, -1	0	NT	
0x003C		bnez x4, L0	0	NT	
0x0040		...	0	NT	

Iteration 1

aa = 4
bb = 3

address	Instruction		BHT	Prediction	misP. counter
0x0000	L0:	...	0	NT	
...		...	0	NT	
0x0010		addi x3, x1, -2	0	NT	
0x0014		bnez x3, L1	1	NT	1
0x0018		add x1, x0, x0	0	NT	
0x001C	L1:	addi x3, x2, -2	0	NT	
0x0020		bnez x3, L2	1	NT	1
0x0024		add x2, x0, x0	0	NT	
0x0028	L2:	sub x3, x1, x2	0	NT	
0x002C		bnez x3, L3	1	NT	1
0x0030		...	0	NT	
0x0034	L3:	...	0	NT	
0x0038		addi x4, x4, -1	0	NT	
0x003C		bnez x4, L0	1	NT	1
0x0040		...	0	NT	

Iteration 2

aa = 4
bb = 3

address	Instruction		BHT	Prediction	misP. counter
0x0000	L0:	...	0	NT	
...		...	0	NT	
0x0010		addi x3, x1, -2	0	NT	
0x0014		bnez x3, L1	2	NT	2
0x0018		add x1, x0, x0	0	NT	
0x001C	L1:	addi x3, x2, -2	0	NT	
0x0020		bnez x3, L2	2	NT	2
0x0024		add x2, x0, x0	0	NT	
0x0028	L2:	sub x3, x1, x2	0	NT	
0x002C		bnez x3, L3	2	NT	2
0x0030		...	0	NT	
0x0034	L3:	...	0	NT	
0x0038		addi x4, x4, -1	0	NT	
0x003C		bnez x4, L0	2	NT	2
0x0040		...	0	NT	

Iteration 3

aa = 4
bb = 3

address	Instruction		BHT	Prediction	misP. counter
0x0000	L0:	...	0	NT	
...		...	0	NT	
0x0010		addi x3, x1, -2	0	NT	
0x0014		bnez x3, L1	3	T	2
0x0018		add x1, x0, x0	0	NT	
0x001C	L1:	addi x3, x2, -2	0	NT	
0x0020		bnez x3, L2	3	T	2
0x0024		add x2, x0, x0	0	NT	
0x0028	L2:	sub x3, x1, x2	0	NT	
0x002C		bnez x3, L3	3	T	2
0x0030		...	0	NT	
0x0034	L3:	...	0	NT	
0x0038		addi x4, x4, -1	0	NT	
0x003C		bnez x4, L0	3	T	2
0x0040		...	0	NT	

Iteration 4

aa = 4
bb = 3

address	Instruction		BHT	Prediction	misP. counter
0x0000	L0:	...	0	NT	
...		...	0	NT	
0x0010		addi x3, x1, -2	0	NT	
0x0014		bnez x3, L1	3	T	2
0x0018		add x1, x0, x0	0	NT	
0x001C	L1:	addi x3, x2, -2	0	NT	
0x0020		bnez x3, L2	3	T	2
0x0024		add x2, x0, x0	0	NT	
0x0028	L2:	sub x3, x1, x2	0	NT	
0x002C		bnez x3, L3	3	T	2
0x0030		...	0	NT	
0x0034	L3:	...	0	NT	
0x0038		addi x4, x4, -1	0	NT	
0x003C		bnez x4, L0	3	T	2
0x0040		...	0	NT	

Iteration 100

aa = 4
bb = 3

address	Instruction		BHT	Prediction	misP. counter
0x0000	L0:	...	0	NT	
...		...	0	NT	
0x0010		addi x3, x1, -2	0	NT	
0x0014		bnez x3, L1	3	T	2
0x0018		add x1, x0, x0	0	NT	
0x001C	L1:	addi x3, x2, -2	0	NT	
0x0020		bnez x3, L2	3	T	2
0x0024		add x2, x0, x0	0	NT	
0x0028	L2:	sub x3, x1, x2	0	NT	
0x002C		bnez x3, L3	3	T	2
0x0030		...	0	NT	
0x0034	L3:	...	0	NT	
0x0038		addi x4, x4, -1	0	NT	
0x003C		bnez x4, L0	2	T	3
0x0040		...	0	NT	

BHT Example

Considering a 2-bit saturating counter BHT of 1K entries; and assuming that the processor executes the following code fragment, determine the BHT final state and calculate the misprediction ratio for the presented case.

Case 1:

-Program inputs for aa and bb are always different than 2.

Mispredition ratio =

Number of mispredicted branches / Total branches

Mispredition ratio = 9 / 400

Mispredition ratio = 2.25%

BHT Example - Case 2

Considering a 2-bit saturating counter BHT of 1K entries; and assuming that the processor executes the following code fragment, determine the BHT final state and calculate the misprediction ratio for the presented case.

Case 2:

-Program inputs for aa and bb alternate values different than 2 and equal to 2.

Case 2 - Iteration 1

aa = 4
bb = 3

address	Instruction		BHT	Prediction	misP. counter
0x0000	L0:	...	0	NT	
...		...	0	NT	
0x0010		addi x3, x1, -2	0	NT	
0x0014		bnez x3, L1	0	NT	
0x0018		add x1, x0, x0	0	NT	
0x001C	L1:	addi x3, x2, -2	0	NT	
0x0020		bnez x3, L2	0	NT	
0x0024		add x2, x0, x0	0	NT	
0x0028	L2:	sub x3, x1, x2	0	NT	
0x002C		bnez x3, L3	0	NT	
0x0030		...	0	NT	
0x0034	L3:	...	0	NT	
0x0038		addi x4, x4, -1	0	NT	
0x003C		bnez x4, L0	0	NT	
0x0040		...	0	NT	

Case 2 - Iteration 1

aa = 4
bb = 3

address	Instruction	BHT	Prediction	misP. counter
0x0000	L0: ...	0	NT	
...	...	0	NT	
0x0010	addi x3, x1, -2	0	NT	
0x0014	bnez x3, L1	1	NT	1
0x0018	add x1, x0, x0	0	NT	
0x001C	L1: addi x3, x2, -2	0	NT	
0x0020	bnez x3, L2	1	NT	1
0x0024	add x2, x0, x0	0	NT	
0x0028	L2: sub x3, x1, x2	0	NT	
0x002C	bnez x3, L3	1	NT	1
0x0030	...	0	NT	
0x0034	L3: ...	0	NT	
0x0038	addi x4, x4, -1	0	NT	
0x003C	bnez x4, L0	1	NT	1
0x0040	...	0	NT	

Case 2 - Iteration 2

aa = 2
bb = 2

address	Instruction	BHT	Prediction	misP. counter
0x0000	L0: ...	0	NT	
...	...	0	NT	
0x0010	addi x3, x1, -2	0	NT	
0x0014	bnez x3, L1	0	NT	1
0x0018	add x1, x0, x0	0	NT	
0x001C	L1: addi x3, x2, -2	0	NT	
0x0020	bnez x3, L2	0	NT	1
0x0024	add x2, x0, x0	0	NT	
0x0028	L2: sub x3, x1, x2	0	NT	
0x002C	bnez x3, L3	0	NT	1
0x0030	...	0	NT	
0x0034	L3: ...	0	NT	
0x0038	addi x4, x4, -1	0	NT	
0x003C	bnez x4, L0	2	NT	2
0x0040	...	0	NT	

Case 2 - Iteration 3

aa = 4
bb = 3

address	Instruction		BHT	Prediction	misP. counter
0x0000	L0:	...	0	NT	
...		...	0	NT	
0x0010		addi x3, x1, -2	0	NT	
0x0014		bnez x3, L1	1	NT	2
0x0018		add x1, x0, x0	0	NT	
0x001C	L1:	addi x3, x2, -2	0	NT	
0x0020		bnez x3, L2	1	NT	2
0x0024		add x2, x0, x0	0	NT	
0x0028	L2:	sub x3, x1, x2	0	NT	
0x002C		bnez x3, L3	1	NT	2
0x0030		...	0	NT	
0x0034	L3:	...	0	NT	
0x0038		addi x4, x4, -1	0	NT	
0x003C		bnez x4, L0	3	T	2
0x0040		...	0	NT	

Case 2 - Iteration 4

aa = 2
bb = 2

address	Instruction	BHT	Prediction	misP. counter
0x0000	L0: ...	0	NT	
...	...	0	NT	
0x0010	addi x3, x1, -2	0	NT	
0x0014	bnez x3, L1	0	NT	2
0x0018	add x1, x0, x0	0	NT	
0x001C	L1: addi x3, x2, -2	0	NT	
0x0020	bnez x3, L2	0	NT	2
0x0024	add x2, x0, x0	0	NT	
0x0028	L2: sub x3, x1, x2	0	NT	
0x002C	bnez x3, L3	0	NT	2
0x0030	...	0	NT	
0x0034	L3: ...	0	NT	
0x0038	addi x4, x4, -1	0	NT	
0x003C	bnez x4, L0	3	T	2
0x0040	...	0	NT	

Case 2 - Iteration 100

aa = 2
bb = 2

address	Instruction		BHT	Prediction	misP. counter
0x0000	L0:	...	0	NT	
...		...	0	NT	
0x0010		addi x3, x1, -2	0	NT	
0x0014		bnez x3, L1	0	NT	50
0x0018		add x1, x0, x0	0	NT	
0x001C	L1:	addi x3, x2, -2	0	NT	
0x0020		bnez x3, L2	0	NT	50
0x0024		add x2, x0, x0	0	NT	
0x0028	L2:	sub x3, x1, x2	0	NT	
0x002C		bnez x3, L3	0	NT	50
0x0030		...	0	NT	
0x0034	L3:	...	0	NT	
0x0038		addi x4, x4, -1	0	NT	
0x003C		bnez x4, L0	2	T	3
0x0040		...	0	NT	

BHT Example - Case 2

Considering a 2-bit saturating counter BHT of 1K entries; and assuming that the processor executes the following code fragment, determine the BHT final state and calculate the misprediction ratio for the presented case.

Case 2:

-Program inputs for aa and bb alternate values different than 2 and equal to 2.

Mispredition ratio =

Number of mispredicted branches / Total branches

Mispredition ratio = 153 / 400

Mispredition ratio = 38.25%

(2,2) Example

Considering a (2,2) correlating predictor of 1K entries; and assuming that the processor executes the following code fragment, determine the (2,2) final state and calculate the misprediction ratio for the presented case.

	L0:	...
for (i=1;i<=100;i++) {		addi x3, x1, -2
read_data();		bnez x3, L1
if (aa==2)		add x1, x0, x0
aa = 0;	L1:	addi x3, x2, -2
if (bb==2)		bnez x3, L2
bb = 0;	L2:	add x2, x0, x0
if (aa == bb)		sub x3, x1, x2
{...}		bnez x3, L3
}	L3:	...
		...
		addi x4, x4, -1
		bnez x4, L0
		...

(2,2) Example

Considering a (2,2) correlating predictor of 1K entries; and assuming that the processor executes the following code fragment, determine the (2,2) final state and calculate the misprediction ratio for the presented case.

Case 1:

-Program inputs for aa and bb alternate values different than 2 and equal to 2.

(2,2) - Iteration 1

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	00		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	0	0	0	0			
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	0	0	0			
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	0			
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	0	0	0	0			
0x0040	...	0	0	0	0			

(2,2) - Iteration 1

aa = 3
bb = 4

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	00		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	0	0	0	0	00		
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	0	0	0			
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	0			
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	0	0	0	0			
0x0040	...	0	0	0	0			

(2,2) - Iteration 1

aa = 3
bb = 4

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	00		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	0	0	0	00		1
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	0	0	0			
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	0			
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	0	0	0	0			
0x0040	...	0	0	0	0			

(2,2) - Iteration 1

aa = 3
bb = 4

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	00		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	0	0	0	00	01	1
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	0	0	0			
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	0			
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	0	0	0	0			
0x0040	...	0	0	0	0			

(2,2) - Iteration 1

aa = 3
bb = 4

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	00		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	0	0	0	00	01	1
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	1	0	0	01	11	1
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	0			
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	0	0	0	0			
0x0040	...	0	0	0	0			

(2,2) - Iteration 1

aa = 3
bb = 4

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	00		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	0	0	0	00	01	1
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	1	0	0	01	11	1
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	1	11	11	1
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	0	0	0	0			
0x0040	...	0	0	0	0			

(2,2) - Iteration 1

aa = 3
bb = 4

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	00		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	0	0	0	00	01	1
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	1	0	0	01	11	1
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	1	11	11	1
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	0	0	0	1	11	11	1
0x0040	...	0	0	0	0			

(2,2) - Iteration 2

aa = 2
bb = 2

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	11		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	0	0	0	11	10	1
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	1	0	0	10	00	1
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	1	00	00	1
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	1	0	0	1	00	01	2
0x0040	...	0	0	0	0			

(2,2) - Iteration 3

aa = 3
bb = 4

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	01		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	1	0	0	01	11	2
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	1	0	1	11	11	2
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	2	11	11	2
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	1	0	0	2	11	11	3
0x0040	...	0	0	0	0			

(2,2) - Iteration 4

aa = 2
bb = 2

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	11		
...	...	0	0	0	0	init	End	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	1	0	0	11	10	2
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	1	0	1	10	00	2
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	2	00	00	2
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	2	0	0	2	00	01	4
0x0040	...	0	0	0	0			

(2,2) - Iteration 5

aa = 3
bb = 4

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	01		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	2	0	0	01	11	3
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	1	0	2	11	11	3
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	3	11	11	2
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	2	0	0	3	11	11	4
0x0040	...	0	0	0	0			

(2,2) - Iteration 6

aa = 2
bb = 2

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	11		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	2	0	0	11	10	3
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	1	0	2	10	00	3
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	3	00	00	2
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	3	0	0	3	00	01	4
0x0040	...	0	0	0	0			

(2,2) - Iteration 7

aa = 3
bb = 4

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	01		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	3	0	0	01	11	3
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	1	0	3	11	11	3
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	3	11	11	2
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	3	0	0	3	11	11	4
0x0040	...	0	0	0	0			

(2,2) - Iteration 8

aa = 2
bb = 2

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	11		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	3	0	0	11	10	3
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	1	0	3	10	00	3
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	3	00	00	2
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	3	0	0	3	00	01	4
0x0040	...	0	0	0	0			

(2,2) - Iteration 100

aa = 2
bb = 2

address	Instruction	00	01	10	11	2-b sft reg		misP. counter
0x0000	L0: ...	0	0	0	0	11		
...	...	0	0	0	0	init	end	
0x0010	addi x3, x1, -2	0	0	0	0			
0x0014	bnez x3, L1	1	3	0	0	11	10	3
0x0018	add x1, x0, x0	0	0	0	0			
0x001C	L1: addi x3, x2, -2	0	0	0	0			
0x0020	bnez x3, L2	0	1	0	3	10	00	3
0x0024	add x2, x0, x0	0	0	0	0			
0x0028	L2: sub x3, x1, x2	0	0	0	0			
0x002C	bnez x3, L3	0	0	0	3	00	00	2
0x0030	...	0	0	0	0			
0x0034	L3: ...	0	0	0	0			
0x0038	addi x4, x4, -1	0	0	0	0			
0x003C	bnez x4, L0	2	0	0	3	00	00	5
0x0040	...	0	0	0	0			

(2,2) Example

Considering a (2,2) correlating predictor of 1K entries; and assuming that the processor executes the following code fragment, determine the (2,2) final state and calculate the misprediction ratio for the presented case.

Case 1:

-Program inputs for aa and bb alternate values different than 2 and equal to 2.

Misprediction ratio =

Number of mispredicted branches / Total branches

Mispredition ratio = 13 / 400

Mispredition ratio = 3.25%