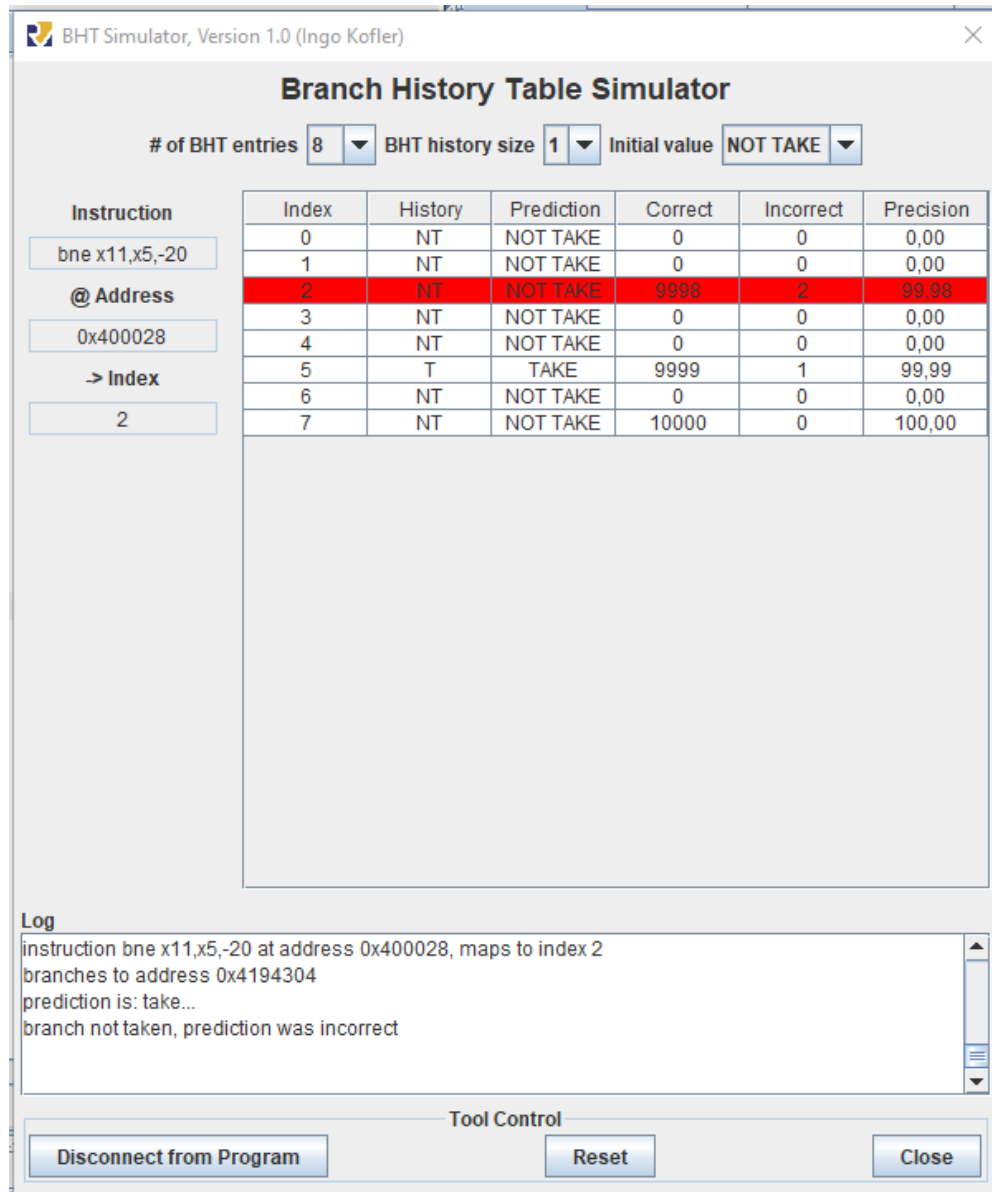


1 Формальный отчет

1. Выполнил: Комиссаров Данил Андреевич.
2. Студент группы Б01-304.
3. Первый вариант задания.
4. Контакты: komissarov.da@phystech.edu
- 5.



1. Точность **bagic_br_1** составляет 99,99%.
Точность **bagic_br_2** составляет 100%.

Branch History Table Simulator

of BHT entries BHT history size Initial value

Instruction

bne x11,x5,-44

@ Address

0x400040

-> Index

0

Index	History	Prediction	Correct	Incorrect	Precision
0	NT	NOT TAKE	9998	2	99,98
1	NT	NOT TAKE	0	0	0,00
2	NT	NOT TAKE	0	0	0,00
3	NT	NOT TAKE	0	0	0,00
4	NT	NOT TAKE	0	0	0,00
5	NT	NOT TAKE	0	20000	0,00
6	NT	NOT TAKE	0	0	0,00
7	NT	NOT TAKE	0	0	0,00

Log

instruction bne x11,x5,-44 at address 0x400040, maps to index 0
 branches to address 0x4194280
 prediction is: take...
 branch not taken, prediction was incorrect

Tool Control

Disconnect from Program

Reset

Close

```

1  .text
2  main:
3      li    t0, 10000
4      li    a3, 1
5      li    a4, 0
6      li    a1, 0
7  loop:
8      beq    a4, zero, magic_br_1 # branch #1
9      nop
10     nop
11     nop
12     nop
13     nop
14     nop
15     addi   a4, a4, 0
16 magic_br_1:
17     beq    a3, zero, magic_br_2 # branch #2
18     addi   a3, a3, 0
19 magic_br_2:
20
21     # ***** ADD HERE *****
22     # your code for task 4
23     # *****
24
25     addi   a1, a1, 1
26     bne    a1, t0, loop
27

```

2. Индексы модулей предсказания для **beq branch1** и **branch2** совпали, а так как в настройках предиктора выставлен конечный автомат с двумя состояниями, то обманывать его можно, попеременно выдавая результаты *taken* - *not taken*. Так и добиваемся точности работы предиктора равной нулю.

BHT Simulator, Version 1.0 (Ingo Kofler)

Branch History Table Simulator

of BHT entries BHT history size Initial value

Index	History	Prediction	Correct	Incorrect	Precision
0	T, NT	TAKE	9997	3	99.97
1	NT, NT	NOT TAKE	0	0	0,00
2	NT, NT	NOT TAKE	0	0	0,00
3	NT, NT	NOT TAKE	0	0	0,00
4	NT, NT	NOT TAKE	0	0	0,00
5	T, NT	NOT TAKE	10000	10000	50,00
6	NT, NT	NOT TAKE	0	0	0,00
7	NT, NT	NOT TAKE	0	0	0,00

Instruction:
 @ Address:
 -> Index:

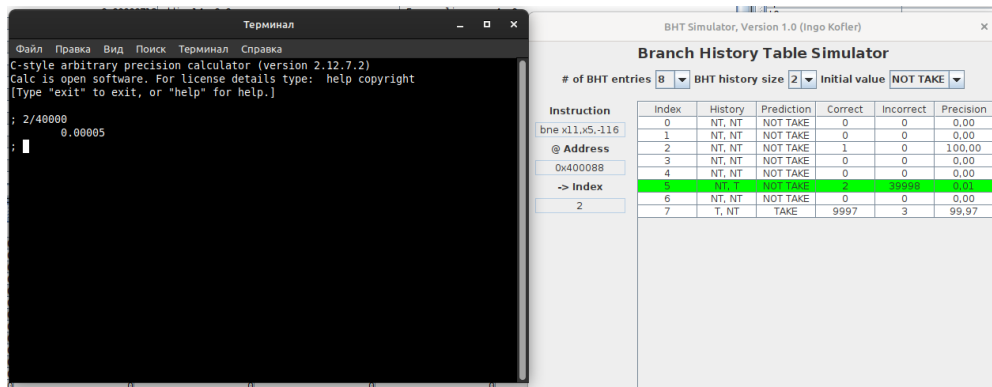
Log

```

instruction bne x11,x5,-44 at address 0x400040, maps to index 0
branches to address 0x4194280
prediction is: take...
branch not taken, prediction was incorrect
  
```

Tool Control

3. В отличие от предиктора из первого задания, сейчас установлен предиктор с 4 состояниями, и причем он построен так, чтобы вместо **not taken** выдавать **taken**, необходимо ошибиться 2 раза, что и происходит в цикле *loop*. Так этот предиктор ошибается на 1 раз больше в цикле.



```

8      beq      a4, zero, magic_br_1 # branch #1
9      nop
10     nop
11     nop
12     nop
13     nop
14     nop
15     addi     a4, a4, 0
16 magic_br_1:
17     beq      a3, zero, magic_br_2 # branch #2
18     addi     a3, a3, 0
19 magic_br_2:
20
21 # ***** ADD HERE *****
22     nop
23     nop
24     nop
25     nop
26     nop
27     nop
28     beq      a3, zero, foo_1 # branch #3
29     nop
30     nop
31     nop
32     nop
33     nop
34     nop
35     nop
36 foo_1:
37     beq      a4, zero, foo_2 # branch #4
38 foo_2:
39
40     addi     a1, a1, 1
41     bne      a1, t0, loop
42     addi     a1, a1, -1
43 # *****
44
45     addi     a1, a1, 1
46     bne      a1, t0, loop
47

```

4. Действительно, так и сделаем: цикл **taken - taken - not taken - not taken**. В конце концов на ассемблере пишем, даже внутри ограниченного пространства можем перелопатить программу до неузнаваемости. Корректно процент предсказания уже не прописывается в программе, считаем отдельно в калькуляторе. Увеличивая начальное значение **t0**, уменьшается процент предсказания, но я не стал увеличивать, так как комп не тянет.