

# Grafický simulátor superskalárních procesorů s webovým rozhraním

Bc. Michal Majer

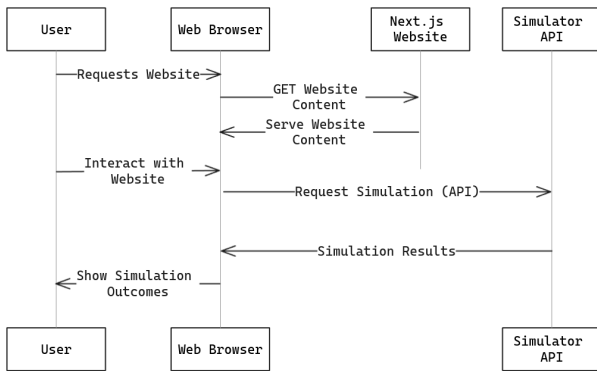
Vedoucí: doc. Ing. Jiří Jaroš, Ph.D.



28. ledna 2024

- Webový **simulátor** superskalárního procesoru
- Vylepšení simulace
- Užitečný nástroj pro výuku

- Simulační rozhraní (HTTP a CLI)
- Webová aplikace v Reactu
- Vylepšení implementace simulátoru
- Důkladné testování



- HTTP API
- Funkční webová aplikace schopná simulovat
- Integrace s GCC
- Kontejnerizace
- Zásadní změny v mnoha modulech simulátoru
  - Syntax a popis instrukcí
  - Skoky
  - Paměťový modul
  - Statistiky
  - ...

### Program

```

add:
0x0  addi sp,sp,-48
0x4  sw s8,44,sp
0x8  addi s8,sp,48
0xc  sw a0,-36,s8
0x10 sw a1,-48,s8
0x14 lw a5,-36,s8
0x18 addi a5,a5,1
0x1c slli a5,a5,1
0x20 sw a5,-28,s8
0x24 lw a4,-28,s8
0x28 lw a5,-48,s8
0x2c add a5,a4,a5
0x30 sw a5,-28,s8
0x34 sw zero,-24,s8
0x38
        
```

### Branch Block

No relevant entries

### Fetch Block

PC: 208

nop  
 nop  
 nop

### Decode Block

Stalled: No

empty  
 empty  
 empty

### Reorder Buffer

Capacity: 15/256

sw tg4,-28(tg1)  
 lw tg5,-28(tg1)  
 lw tg6,-48(tg1)  
 add tg7,tg5,tg6  
 sw tg7,-20(tg1)  
 sw zero,-24(tg1)  
 j .L2  
 lw tg8,-24(tg1) S  
 lw tg9,-36(tg1) S  
 blt tg8,tg9,.L3 S  
 lw tg10,-20(tg1) S  
 mv tg11,tg10 S  
 lw tg12,44(tg0) S  
 addi tg13,tg0,48 S  
 jr ra S  
 empty

### ALU Issue Window

1 instruction

Instruction	Arg 1	Arg 2
mv tg11,tg10	0	
empty		
empty		
empty		
empty		
empty		

### FX Universal

0/1

empty

### FP Issue Window

0 instructions

Instruction	Arg 1	Arg 2
empty		
empty		
empty		
empty		
empty		
empty		

### FP

0/0

empty

### Branch Issue Window

1 instruction

Instruction	Arg 1	Arg 2
blt tg8,tg9,.L3	0	-36
empty		
empty		
empty		
empty		
empty		

### Branch

0/2

empty

### Memory

2/1

sw tg4,-28(tg1)

### Main Memory

0x0080 00 00 00 00 00 00 00 00  
 0x0088 00 00 00 00 00 00 00 00  
 0x0010 00 00 00 00 00 00 00 00  
 0x0018 00 00 00 00 00 00 00 00  
 0x0078 00 00 00 00 00 00 00 00

### Cache

LRU

Index	Tag	Line
0x00	0x00	00 00
0x00	0x00	00 00

### Store Buffer

Load Buffer

The screenshot displays the MIPS simulator interface during Lab 10. The top section shows the assembly code editor with the following instructions:

```

a001 tg1s, t0s, 48
jr ra
sw tg4, -28(tg1)
sw zero, -24(tg1)

```

Below the code editor, the 'Memory' panel shows the current memory address as 2/1. The 'Store Buffer' and 'Load Buffer' panels are empty, indicating no pending store or load instructions. The 'Main Memory' panel shows a large table of memory addresses and their contents, with some entries highlighted in green. The 'Cache' panel on the right shows the LRU cache state, with columns for Index, Tag, and Line. The cache contains entries for addresses 0x00, 0x02, 0x04, 0x06, 0x08, 0x0a, 0x0c, and 0x0e, all with a tag of 0x00.

## ISA Configuration

Active configuration

Isa 1



Save Changes

No memory locations defined.

Name	Buffers	Functional Units	Cache	Memory	Branch
<h3>Cache</h3> <p><input checked="" type="checkbox"/> Use cache</p> <p>Cache lines ⓘ 16</p> <p>Cache line size (B) ⓘ 32</p> <p>Cache associativity ⓘ 2</p> <div> <div> <p>Cache replacement policy</p> <p>LRU FIFO Random</p> </div> <div> <p>Store behavior</p> <p>write-back write-through</p> </div> </div> <p>Lane replacement delay ⓘ 10</p> <p>Cache access delay ⓘ</p>					

## Code Editor

C

ASM

Examples

☒ Optimize
 ☒ Rename registers
 ☐ Unroll loops
 ☐ Peel loops
 ☒ Inline functions
 ☒ Omit frame pointer

Compile

C Code

Load

Save

```

1  int array [16] = {5, 6, 7, 1, 2, 1, 8, 4,
2      8, 4, 3, 9, 5, 5, 6, 7};
3
4  int main()
5  {
6      int n = 16;
7      int temp;
8      for (int i = 0; i < n - 1; i = i+1) {
9          for (int j = 0; j < n - i - 1; j = j+1) {
10             if (array[j] > array[j + 1]){
11                 temp = array[j];
12                 array[j] = array[j + 1];
13                 array[j + 1] = temp;
14             }
15         }
16     }
17 }
18
19

```

ASM Code

Check

Load

Save

```

1  main:
2      lla a1,array
3      lla a2,array+60
4      .L2:
5          mv a5,a1
6      .L4:
7          lw a4,0(a5)
8          lw a3,4(a5)
9          ble a4,a3,.L3
10         sw a3,0(a5)
11         sw a4,4(a5)
12     .L3:
13         addi a5,a5,4
14         bne a5,a2,.L4
15         addi a2,a5,-4
16         bne a2,a1,.L2
17         li a0,0
18         ret
19         .align 2
20     array:
21         .word 5
22         .word 6
23         .word 7
24         .word 1
25         .word 2
26         .word 1
27         .word 8
28         .word 4
29         .word 8
30         .word 4
31         .word 3
32         .word 9
33         .word 5
34         .word 5
35         .word 6
36         .word 7

```



## Memory Editor

Array1

Array2

New

Pointer Name ⓘ

Array3

Data type ⓘ

Byte

Short

Integer

Unsigned Integer

Long

Unsigned Long

Float

Double

Boolean

Char

Byte

Short

Alignment (Bytes) ⓘ

16

Data Source

A Constant

Random Numbers

CSV File

### Data Options

Random data will be generated after each time the memory location is saved.

Lower bound of random range

0

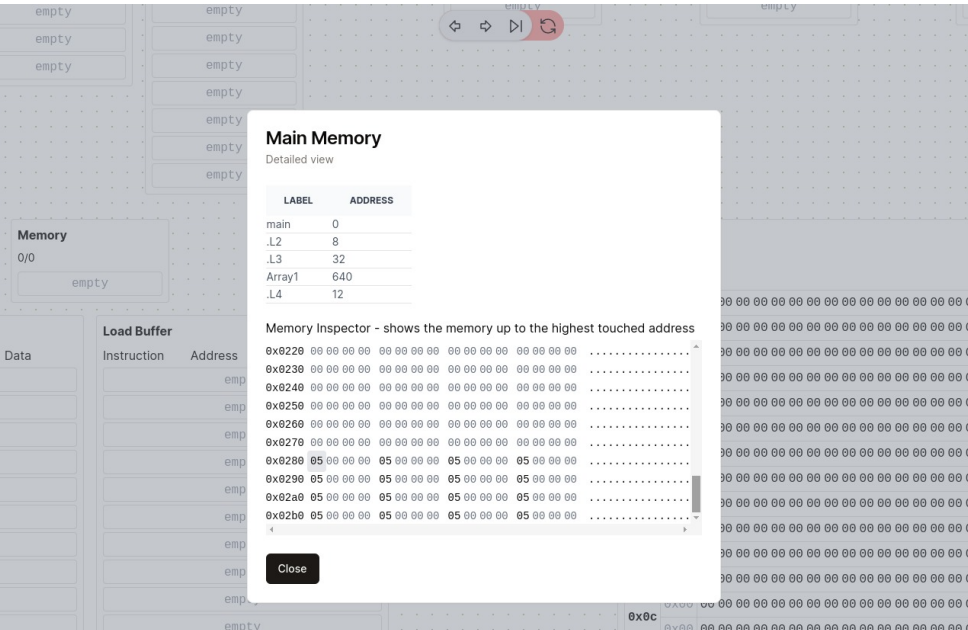
Upper bound of random range

2

Number of Elements

8

Create



## Main Memory

Detailed view

LABEL	ADDRESS
main	0
.L2	8
.L3	32
Array1	640
.L4	12

Memory Inspector - shows the memory up to the highest touched address

0x0220	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....
0x0230	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....
0x0240	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....
0x0250	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....
0x0260	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....
0x0270	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	.....
0x0280	05 00 00 00	05 00 00 00	05 00 00 00	05 00 00 00	.....
0x0290	05 00 00 00	05 00 00 00	05 00 00 00	05 00 00 00	.....
0x02a0	05 00 00 00	05 00 00 00	05 00 00 00	05 00 00 00	.....
0x02b0	05 00 00 00	05 00 00 00	05 00 00 00	05 00 00 00	.....

Close

## Simulation Statistics

IPC

0.31

Clocks

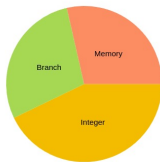
26

Branch Prediction Accuracy

0.00%

### Static Instruction Mix

Instruction mix of the program



Table

Chart

### Dynamic Instruction Mix

Instruction mix of committed instructions

Category	Value	Proportion
Integer	4	50.00%
Float	0	0.00%
Branch	2	25.00%
Memory	2	25.00%
Other	0	0.00%

Table

Chart

### Functional Units

Amount of time a unit spends computing

Name	Busy Cycles	Utilization
L_S	4	15.38%
Branch	11	42.31%
Memory	5	19.23%
FX Universal	13	50.00%
FP	0	0.00%

### Detailed Statistics

### Cache Statistics

### Instruction Statistics

Heat Map of execution counts

Datum	Popis
5.2.	Běhové statistiky
12.2.	Ladící výstupy
19.2.	CLI
26.2.	Chybějící části simulačního náhledu – pole registrů, prediktor
4.3.	Chybějící části simulačního náhledu – detaily bloků, výjimky
11.3.	Testování - logika simulátoru
18.3.	Optimalizace výkonu simulátoru i webové aplikace, implementace benchmarků
25.3.	Design aplikace, detaily uživatelského rozhraní, výukové texty aplikace
1.4.	Uživatelské testování, zapracování připomínek
8.4.	Testování uživatelského rozhraní
15.4.	Stabilizace implementace, dokumentace
22.4.	Měření, vyhodnocení výsledků
29.4.	Text práce
6.5.	Text práce

