

INIAD Computer Architecture Practice

# 第4回 コード実行をシミュレーションする

## Simulate code execution

# 1. この実習の概要

## Outline of this course

## このコースの目標 Goal of this course

- CPUのマシンコードを書いて実行する  
Write and execute CPU machine code.
  - CPUの命令セットを確認する  
Check the CPU instruction set.
  - 命令のシミュレータを作る  
Make an instruction simulator.
  - 命令コードの簡易アセンブラを作る  
Make a simple assembler of instruction code.
  - マシンコードを書いてそのコードの実行をシミュレーションする  
Write machine code and simulate the execution of the code.

# 開発対象の命令セット Instruction set to be developed

## ● M32R命令セット M32R instruction set

### ■ CA講座のCPU16命令セットの出典

It is the source of the CPU16 instruction set in CA course.

- CPU16は16ビットCPUだったが、M32Rは32ビットCPU  
CPU16 was a 16-bit CPU, but M32R is a 32-bit CPU.

### ■ 参考資料 Reference documents

- 日本語:  
[https://www.renesas.com/jp/ja/doc/products/mpumcu/002/rjj09b0107\\_32fpusm.pdf](https://www.renesas.com/jp/ja/doc/products/mpumcu/002/rjj09b0107_32fpusm.pdf)
- English:  
<https://1jzmerc.com/files/pdf/mcu/m3217x-e32fpu.pdf>

## 2. 前回の課題から

# From the previous challenge

# 前回の課題 The previous challenge

- asm32r開発を完成させる Complete the development of asm32r
  - 提供された元のバージョン Original version provided:
    - 元のバージョンは、表 “isa32r.xlsx” の太字で示した命令を処理することができる。  
The original version can process the instructions shown in bold in the table “isa32r.xlsx”.
  - やって欲しいこと What you should do:
    - 表 “isa32r.xlsx” 中の太字以外の命令の処理を追加してアセンブラを完成すること。  
Complete the assembler by adding instructions other than bold in the table “isa32r.xlsx”.

# 前回の課題 The previous challenge

## ● アセンブラ asm32r.c について About assembler asm32r.c

### ■ 使い方 Usage:

- > asm32r "-" (省略可能)  
> asm32r "-" (optional)
- アセンブリ言語で書かれたソースコードは、ファイル"asmcode.txt"から入力する。  
Source code written in assembly language is input from the file "asmcode.txt".
- "-" : "-"が付いていると命令名を出力しない。  
"-": When "-" is added, the instruction names are not output.

## 前回の課題 The previous challenge

- サンプルコード asmcode3.txt について

### About sample code asmcode3.txt

- このソースコードは、追加する命令の使用例を集めたもの。

This source code is a collection of examples of instructions to be added.

- 生成されたコードを確認するためだけのもので、機能的な意味はない。

It is only for checking the generated code and has no functional meaning.



# コード生成関数 Code generation function

- rr() 

op1	d	op2	s
-----	---	-----	---
- ri() 

op1	d	imm8
-----	---	------
- bc() 

op1	op2	pcdisp8
-----	-----	---------
- bc24() 

op1	d	pcdisp24	pcdisp24
-----	---	----------	----------

 \* Pcdisp24 is PC relative and shifted value.
- beq() 

op1	d	op2	s	pcdisp16
-----	---	-----	---	----------

 \* Pcdisp16 is PC relative and shifted value.
- rrd() 

op1	d	op2	s	imm16 / disp16
-----	---	-----	---	----------------
- ld24() 

op1	d	imm24	imm24
-----	---	-------	-------
- ri5() 

op1	d	op2	imm5
-----	---	-----	------

### 3. 命令シミュレータのテストケース

Test cases for the instruction simulator

# テストケース Test case

- 「テストケース」とは What is a "test case"?
  - ある場合について、テストする対象が正しく動作することを確認するためのコード

A test case is a code for verifying that the subject being tested works correctly in some cases.

# テストケース Test case

## ● 命令のテストケースの構成 Instruction test case configuration

### ■ テストケースの仕様 Test case specification

- どんな場合についてテストするかをコメント Comment on when to test.

### ■ 初期設定 Initial setting

- テスト対象の命令を実行するための、リソースや環境の初期設定  
Initial setting of resources and environment to execute the instruction under test.

### ■ テスト対象の命令の実行 Executing the instruction being tested

- テスト対象の命令をシミュレーションする Simulate the instruction under test.

### ■ 期待値の確認 Confirmation of expected values

- 変化すべきリソースの値が正しく変化していることを確認  
Confirm that the value of the resource to change is changing correctly.
- 命令実行の影響を受けないリソースの値が変化しないことを確認  
Confirm that the value of the resource that is not affected by the instruction execution does not change.

### ■ テスト制御 Test control

- テストケース実行がOKなら、次のテストケース実行に移る  
If test case execution is OK, move to next test case execution.
- テストケース実行がNGなら、テストを中断してエラー情報を出力  
If test case execution is NG, abort the test and output error information.

# テストケース Test case

## ● 例 : CMPのテストケース Example: CMP test case

```
// Test case: CMP +,+ // 2オペランドが両方正 Two operands are both positive.
LDI16(R12,0x0001); // テスト番号をR12にセット Set test number to R12.
LDI(R0,1); // 初期設定 Initial setting.
LDI(R1,2); // 初期設定 Initial setting.
CMP(R0,R1); // 命令の実行 Execute the instruction.
BNC24(L_NG); // Cビットの確認 Confirmation of the c-bit value.
CMP(R1,R0); // 命令の実行 Execute the instruction.
BC24(L_NG); // Cビットの確認 Confirmation of the c-bit value.
CMP(R1,R1); // 命令の実行 Execute the instruction.
BC24(L_NG); // Cビットの確認 Confirmation of the c-bit value.
```

# テストケース Test case

## ● 例 : BCのテストケース Example: BC test case

// Test case: BC jump forward // 前方ジャンプ Forward jump.

LDI16(R12,0x0004); // テスト番号をR12にセット Set test number to R12.

LDI(R0,1); // 初期設定 Initial setting.

LDI(R1,2); // 初期設定 Initial setting.

CMP(R0,R1); // 初期設定 Initial setting. (The c-bit is set to 1.)

BC(L1); // 命令の実行 Execute the instruction.

BRA24(L\_NG); // 分岐有無の確認 Confirmation of branching or not.

L1 = addr;

# テストケース Test case

## ● 例：テストの終了処理 Example: Test termination

// テストの終了 End of test.

// ここまで到達すればテストは成功 If you reach this point, the test is successful.

BC24(L\_OK); // テスト成功に分岐 Branch to test success.

L\_NG = addr; // テスト失敗 Test failure.

LDI(R13,-1); // 終了状態としてR13に-1をセット Set -1 to R13 as the end state.

BRA(addr); // 無限ループ Infinite loop.

L\_OK = addr; // テスト成功 Test success.

LDI(R13,1); // 終了状態としてR13に1をセット Set 1 to R13 as the end state.

BRA(addr); // 無限ループ Infinite loop.

# テストケース Test case

## ● 例：テストの終了処理 Example: Test termination

// テストの終了 End of test.

// ここまで到達すればテストは成功 If you reach this point, the test is successful.

BC24(L\_OK); // テスト成功に分岐 Branch to test success.

L\_NG = addr; // テスト失敗 Test failure.

LDI(R13,-1); // 終了状態としてR13に-1をセット Set -1 to R13 as the end state.

BRA(addr); // 無限ループ Infinite loop.

L\_OK = addr; // テスト成功 Test success.

LDI(R13,1); // 終了状態としてR13に1をセット Set 1 to R13 as the end state.

BRA(addr); // 無限ループ Infinite loop.



# テストケース Test case

- テストコード例が次のフォルダに置かれている：

A test program example is in the following folder:

- [https://drive.google.com/drive/u/1/folders/1G6\\_Ma4Wy1Id4bjC64KEaXkaKgWDUUJ6q](https://drive.google.com/drive/u/1/folders/1G6_Ma4Wy1Id4bjC64KEaXkaKgWDUUJ6q)
- テストコード Test code: asmcode4.txt

## 4. 提出物 Submission

# 提出物 Submission

- シミュレータ、アセンブラ、テストケースを完成させて、次の3つのファイルをメールで送ってください：

Complete the simulator, assembler, and test case and email the following three files:

- ソースファイル Source files

- sim32r.c
- asm32r.c
- asmcode.txt

- 送付先と期限 Address and deadline

- To: [shimizu008@iniad.org](mailto:shimizu008@iniad.org)
- 11/15（金）まで By 11/15（Fri）

- 補足コメント Supplementary comments

- プログラムの冒頭に指名とIDを記すこと Write your name and ID at the beginning of the programs.
- シミュレータとアセンブラにはすべての命令を実装すること Implement all instructions in the simulator and the assembler.
- コンパイルエラーを無くしておくこと Eliminate compilation errors.
- テストケースをシミュレーションした時に、エラーが無いようにしておくこと  
Make sure there are no errors when simulating the test case.

# 提出物 Submission

## ● テストケースの割り当て Test case assignment

■ 各命令5ケース以上作る Make 5 or more test cases for each instruction.

Instruction name		ldhr (LDH reg rel)	柳瀬 朝陽
beq	須崎 隆也	ldub	横尾 佳大
beqz	濱垣 江里奈	ldubr (LDUB reg rel)	丸山 翔
bgez	鈴木 龍太	lduh	小野 竣也
bgtz	シ ケイイク	lduhr (LDUH reg rel)	木津 芽伊子
bl	LIAO YIKANG	rem	坂田 和希
bl24 (BL 24-bit disp)	上原 未紅	remu	佐藤 滉太
blez	金丸 堯仁	seth	鈴木 健介
bltz	中西 陸	sll3	土屋 晃希
bne	湯本 彩香	slli	成松 怜央
bnez	米田 悠人	sra	吉川 瑠璃子
div	渋谷 洋人	sra3	上野 卓也
divu	富田 萌子	srai	杉田 柊
ldpi (LD post inc)	滝川 海	srl	鈴木 悠馬
ldb	野村 那津	srl3	江口 昂大
ldbr (LDB reg rel)	高橋 正人	srli	塚田 啓太
ldh	中村 芙由	stpi (ST pre inc)	牧野 俊太郎