



Course number: CSC.T341

# コンピュータ論理設計 演習(7) Computer Logic Design Exercise(7)

情報工学系 荒堀喜貴

Yoshitaka ARAHORI, Department of Computer Science  
arahori\_at\_c.titech.ac.jp



Computer Logic Design support page <https://www.arch.cs.titech.ac.jp/lecture/CLD/>

# コンピュータ論理設計 演習(Exercise)の注意点

- 演習はACRiルームを利用します。
- 3~4人のグループを作成します. そのグループ内で情報を共有しながら演習を進めてください.
- 問題はグループ内で相談して解決する, あるいは, 担当のTA(Teaching Assistant)や教員に質問してください.
- 演習には出席点があります. 休まずにきちんと出席しましょう.
- 演習スライドにチェックポイントの図がある場所は, 作業を確認してもらう場所です. すべてのチェックポイントをクリアしましょう.



- 演習時間でなくてもACRiルームを利用できます. 現在は, 1日に4枠(3時間 × 4枠 = 12時間)を利用できます. 独自のハードウェア設計などに挑戦しましょう.

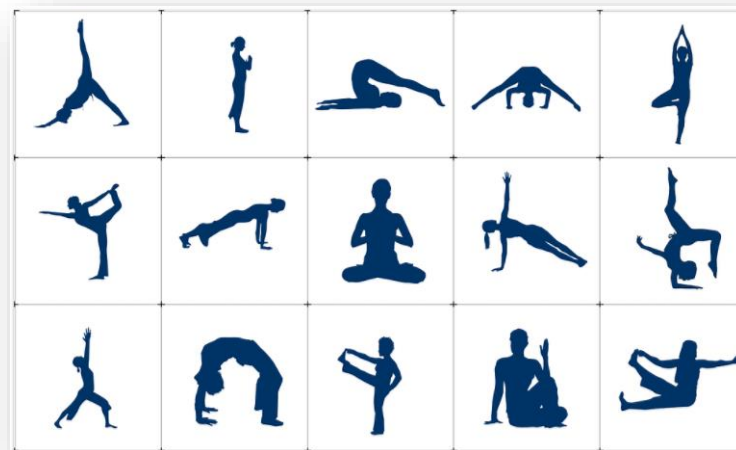


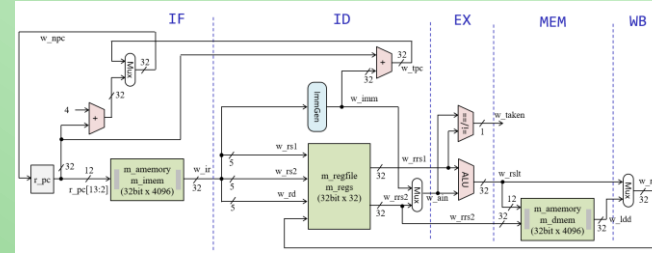
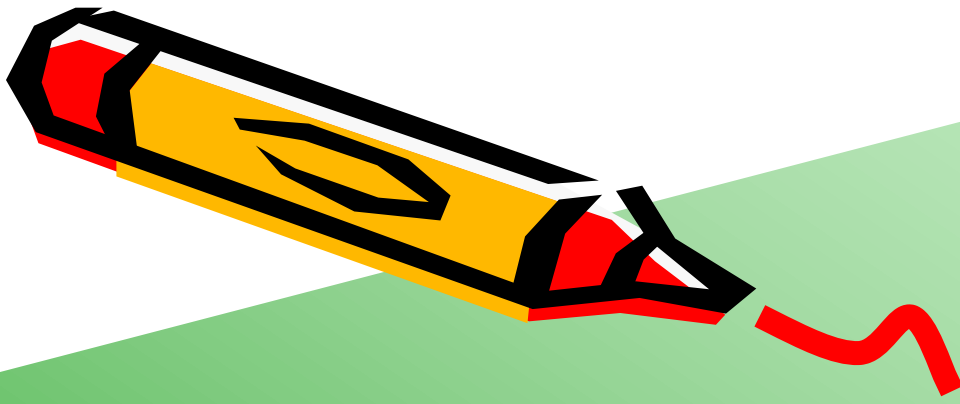
# Exercise(7)

- Project\_8

- プロセッサデザインコンテストのルールなどを理解する.
- プロセッサデザインコンテストの準備を進める.

- チェックポイント (CP) の確認は5月25日(木) 10:30 まで.  
これまでに, できるだけ多くのチェックポイントを通過すること.





# Processor Design Contest



# Design & Implementation

- コンテストに向けて、各グループで
  - 役割の分担をしましょう.
  - スケジュールの確認をしましょう.
  - 設計を進めましょう.



# プログラムを使って実装中のプロセッサの動作を確認

- `add`, `addi` を用いるサンプルプログラム

/home/tu\_kise/cld/2023/baseline/program1.txt

```
initial begin
cm_ram[0]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; // add x0, x0, x0 // NOP
cm_ram[1]={12'd3, 5'd0, 3'b000, 5'd4, 7'b0010011}; // addi x4, x0, 3 // x4 = 3
cm_ram[2]={12'd4, 5'd0, 3'b000, 5'd5, 7'b0010011}; // addi x5, x0, 4 // x5 = 4
cm_ram[3]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; // add x0, x0, x0 // NOP
cm_ram[4]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; // add x0, x0, x0 // NOP
cm_ram[5]={7'd0, 5'd5, 5'd4, 3'b000, 5'd30, 7'b0110011}; // add x30, x4, x5 // led = x4 + x5
cm_ram[6]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; // add x0, x0, x0 // NOP
cm_ram[7]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; // add x0, x0, x0 // NOP
end
```

```
clock: r_pc      w_ir      w_rrs1  w_ain   r_rslt2  r_led
1: 00000000 00000033 00000000 00000000 00000000 00000000
2: 00000004 00300213 00000000 00000003 00000003 00000000
3: 00000008 00400293 00000000 00000004 00000004 00000000
4: 0000000c 00000033 00000000 00000000 00000000 00000000
5: 00000010 00000033 00000000 00000000 00000000 00000000
6: 00000014 00520f33 00000003 00000004 00000007 00000000
7: 00000018 00000033 00000000 00000000 00000000 00000007
```



# プログラムを使って実装中のプロセッサの動作を確認

- `add`, `addi`, `lw`, `sw` を用いるサンプルプログラム

/home/tu\_kise/cld/2023/baseline/program2.txt

```
initial begin
  cm_ram[0]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; //   add  x0, x0, x0 // NOP
  cm_ram[1]={12'd55, 5'd0, 3'b000, 5'd4, 7'b0010011}; //   addi x4, x0, 55 // x4 = 55
  cm_ram[2]={7'd0, 5'd4, 5'd0, 3'b010, 5'd16, 7'b0100011}; //   sw   x4, 16(x0) // m[16] = x4
  cm_ram[3]={12'd16, 5'd0, 3'b010, 5'd7, 7'b0000011}; //   lw   x7, 16(x0) // x7 = m[16]
  cm_ram[4]={12'd9, 5'd0, 3'b000, 5'd2, 7'b0010011}; //   addi x2, x0, 9 // x2 = 9
  cm_ram[5]={7'd0, 5'd2, 5'd7, 3'b000, 5'd3, 7'b0110011}; //   add  x3, x7, x2 // x3 = x7 + x2
  cm_ram[6]={7'd0, 5'd3, 5'd0, 3'b000, 5'd30, 7'b0110011}; //   add  x30, x0, x3 // led = x3
  cm_ram[7]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; //   add  x0, x0, x0 // NOP
end
```

```
clock: r_pc      w_ir      w_rrs1  w_ain   r_rslt2  r_led
1: 00000000 00000033 00000000 00000000 00000000 00000000
2: 00000004 03700213 00000000 00000037 00000037 00000000
3: 00000008 00402823 00000000 00000010 00000010 00000000
4: 0000000c 01002383 00000000 00000010 00000037 00000000
5: 00000010 00900113 00000000 00000009 00000009 00000000
6: 00000014 002381b3 00000037 00000009 00000040 00000000
7: 00000018 00300f33 00000000 00000040 00000040 00000000
8: 0000001c 00000033 00000000 00000000 00000000 00000040
```



# プログラムを使って実装中のプロセッサの動作を確認

- `add`, `addi`, `sll`, `lw`, `sw` を用いるサンプルプログラム

/home/tu\_kise/cld/2023/baseline/program3.txt

```
initial begin
  cm_ram[0]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; // add x0, x0, x0 // NOP
  cm_ram[1]={12'd3, 5'd0, 3'b000, 5'd4, 7'b0010011}; // addi x4, x0, 3 // x4 = 3
  cm_ram[2]={7'd0, 5'd4, 5'd0, 3'b010, 5'd16, 7'b0100011}; // sw x4, 16(x0) // m[16] = x4
  cm_ram[3]={12'd16, 5'd0, 3'b010, 5'd7, 7'b0000011}; // lw x7, 16(x0) // x7 = m[16]
  cm_ram[4]={12'd1, 5'd0, 3'b000, 5'd2, 7'b0010011}; // addi x2, x0, 1 // x2 = 1
  cm_ram[5]={7'd0, 5'd2, 5'd7, 3'b001, 5'd3, 7'b0110011}; // sll x3, x7, x2 // x3 = x7 << x2
  cm_ram[6]={7'd0, 5'd3, 5'd0, 3'b000, 5'd30, 7'b0110011}; // add x30, x0, x3 // led = x3
  cm_ram[7]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; // add x0, x0, x0 // NOP
end
```

```
clock: r_pc      w_ir      w_rrs1  w_ain   r_rslt2 r_led
1: 00000000 00000033 00000000 00000000 00000000 00000000
2: 00000004 00300213 00000000 00000003 00000003 00000000
3: 00000008 00402823 00000000 00000010 00000010 00000000
4: 0000000c 01002383 00000000 00000010 00000003 00000000
5: 00000010 00100113 00000000 00000001 00000001 00000000
6: 00000014 002391b3 00000003 00000001 00000006 00000000
7: 00000018 00300f33 00000000 00000006 00000006 00000000
8: 0000001c 00000033 00000000 00000000 00000000 00000006
```





# プログラムを使って実装中のプロセッサの動作を確認

- `add`, `addi`, `srl`, `lw`, `sw` を用いるサンプルプログラム

/home/tu\_kise/cld/2023/baseline/program4.txt

```
initial begin
  cm_ram[0]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; //   add  x0, x0, x0 // NOP
  cm_ram[1]={12'd8,      5'd0, 3'b000, 5'd4, 7'b0010011}; //   addi x4, x0, 8 // x4 = 8
  cm_ram[2]={7'd0, 5'd4, 5'd0, 3'b010, 5'd16,7'b0100011}; //   sw   x4, 16(x0) // m[16] = x4
  cm_ram[3]={12'd16,     5'd0, 3'b010, 5'd7, 7'b0000011}; //   lw   x7, 16(x0) // x7 = m[16]
  cm_ram[4]={12'd2,      5'd0, 3'b000, 5'd2, 7'b0010011}; //   addi x2, x0, 2 // x2 = 2
  cm_ram[5]={7'd0, 5'd2, 5'd7, 3'b101, 5'd3, 7'b0110011}; //   srl  x3, x7, x2 // x3 = x7 >> x2
  cm_ram[6]={7'd0, 5'd3, 5'd0, 3'b000, 5'd30,7'b0110011}; //   add  x30,x0, x3 // led = x3
  cm_ram[7]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; //   add  x0, x0, x0 // NOP
end
```

```
clock: r_pc      w_ir      w_rrs1  w_ain   r_rslt2  r_led
1: 00000000 00000033 00000000 00000000 00000000 00000000
2: 00000004 00800213 00000000 00000008 00000008 00000000
3: 00000008 00402823 00000000 00000010 00000010 00000000
4: 0000000c 01002383 00000000 00000010 00000008 00000000
5: 00000010 00200113 00000000 00000002 00000002 00000000
6: 00000014 0023d1b3 00000008 00000002 00000002 00000000
7: 00000018 00300f33 00000000 00000002 00000002 00000000
8: 0000001c 00000033 00000000 00000000 00000000 00000002
```



# プログラムを使って実装中のプロセッサの動作を確認

- `add`, `addi`, `bne` を用いるサンプルプログラム

/home/tu\_kise/cld/2023/baseline/program5.txt

```
initial begin
  cm_ram[0]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; // add x0, x0, x0 // NOP
  cm_ram[1]={12'd5, 5'd0, 3'b000, 5'd4, 7'b0010011}; // addi x4, x0, 5 // x4 = 5
  cm_ram[2]={12'd1, 5'd0, 3'b000, 5'd5, 7'b0010011}; // addi x5, x0, 1 // x5 = 1
  cm_ram[3]={12'd1, 5'd5, 3'b000, 5'd5, 7'b0010011}; // L1:addi x5, x5, 1 // x5 = x5 + 1
  cm_ram[4]={7'h7f, 5'd4, 5'd5, 3'b001, 5'b11101, 7'b1100011}; // bne x5, x4, L1 // goto L1 if x5!=x4
  cm_ram[5]={7'd0, 5'd0, 5'd5, 3'b000, 5'd30, 7'b0110011}; // add x30,x5, x0 // led = x5
  cm_ram[6]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; // add x0, x0, x0 // NOP
  cm_ram[7]={7'd0, 5'd0, 5'd0, 3'b000, 5'd0, 7'b0110011}; // add x0, x0, x0 // NOP
end
```

```
clock: r_pc      w_ir      w_rrs1  w_ain   r_rslt2  r_led
  1: 00000000 00000033 00000000 00000000 00000000 00000000
  2: 00000004 00500213 00000000 00000005 00000005 00000000
  3: 00000008 00100293 00000000 00000001 00000001 00000000
  4: 0000000c 00128293 00000001 00000001 00000002 00000000
  5: 00000010 fe429ee3 00000002 ffffffff 20000000 00000000
  6: 0000000c 00128293 00000002 00000001 00000003 00000000
  7: 00000010 fe429ee3 00000003 ffffffff 30000000 00000000
  8: 0000000c 00128293 00000003 00000001 00000004 00000000
  9: 00000010 fe429ee3 00000004 ffffffff 40000000 00000000
 10: 0000000c 00128293 00000004 00000001 00000005 00000000
 11: 00000010 fe429ee3 00000005 ffffffff 50000000 00000000
 12: 00000014 00028f33 00000005 00000000 00000005 00000000
 13: 00000018 00000033 00000000 00000000 00000000 00000005
```



# References



# References

- Computer Logic Design support page
  - <https://www.arch.cs.titech.ac.jp/lecture/CLD/>
- ACRi Room
  - <https://gw.acri.c.titech.ac.jp>
- ACRi Blog
  - <https://www.acri.c.titech.ac.jp/wordpress/>
- 情報工学系計算機室
  - <http://www.csc.titech.ac.jp/>
- Xilinx Vivado Design Suite
  - <https://japan.xilinx.com/products/design-tools/vivado.html>
- Digilent Arty A7-35T
  - <https://reference.digilentinc.com/reference/programmable-logic/arty-a7/start>
- Verilog HDL
  - <https://ja.wikipedia.org/wiki/Verilog>

