

コンピュータ設計

18D8101020L 大木智也

2020/6/30

18D81010202 大木 智也			
課題 2			
		メモリの番地	機械符号 オフセット
1: Loop :	LDM Y	0	3 19
2	JZE DWAR2	2	8 14
3	DCR	4	6
4	STM Y	5	4 19
5	LDM X	7	3 18
6	DCR	9	6
7	STM X	10	4 18
8	JMP Loop	12	7 0
9 DWAR1 :	LDM X	14	5 18
10	PRT	16	1
11	HLT	17	0
12 X :	DC7	18	7
13 Y :	DC5	19	5

課題 3			
Loop :	LDM Y	0 : 3	17
	JZE DWAR2	2 : 8	15
	DCR	4 : 6	
	STM Y	5 : 9	17
	LDM X	7 : 3	16
	DCR	9 : 5	
	STM X	10 : 1	
	JMP Loop	11 : 9	16
	DWAR2 HLT	13 : 7	0
	X: DC = 0	15 : 0	
	Y: DC = 10	16 : 0	
		17 : 10	

課題 1

```
#include <stdio.h>
#include <stdlib.h>
```

```
unsigned char m[256], pc, reg;
```

```
int main() {
    unsigned char instr, n, add;
    /* Machine Language */
    m[0]=2;
    m[1]=10;
    m[2]=1;
    m[3]=6;
    m[4]=8;
```

```

m[5]=8;
m[6]=7;
m[7]=2;
m[8]=0;

```

```

/* virtual machine */
pc=0;
while(1) {
    printf(" pc=%4d reg=%4d\n", pc, reg);
    instr=m[pc]; pc=pc+1;
    switch(instr) {
        case 0: exit(0);
            break;
        case 1: printf("%d\n", reg);
            break;
        case 2: n=m[pc]; pc=pc+1;
            reg=n;
            break;
        case 3: add=m[pc]; pc=pc+1;
            reg=m[add];
            break;
        case 4: add=m[pc]; pc=pc+1;
            m[add]=reg;
            break;
        case 5: reg=reg+1;
            break;
        case 6: reg=reg-1;
            break;
        case 7: add=m[pc]; pc=pc+1;
            pc=add;
            break;
        case 8: add=m[pc]; pc=pc+1;
            if(reg==0){
                pc=add;
            }
            break;
        case 9: add=m[pc]; pc=pc+1;
            if(reg!=0){
                pc=add;
            }
            break;
        case 10: reg=pc;
            break;
        case 11: pc=reg;
            break;
    }
}
}

```

```

/*
10
9
8
7
6
5
4
3
2
1

```

```
*/
```

課題2

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
unsigned char m[256], pc, reg;
```

```
int main() {
```

```
    unsigned char instr,n, add;
```

```
    /* Machine Language */
```

```
    m[0]=3;
```

```
    m[1]=19;
```

```
    m[2]=8;
```

```
    m[3]=14;
```

```
    m[4]=6;
```

```
    m[5]=4;
```

```
    m[6]=19;
```

```
    m[7]=3;
```

```
    m[8]=18;
```

```
    m[9]=6;
```

```
    m[10]=4;
```

```
    m[11]=18;
```

```
    m[12]=7;
```

```
    m[13]=0;
```

```
    m[14]=3;
```

```
    m[15]=18;
```

```
    m[16]=1;
```

```
    m[17]=0;
```

```
    m[18]=7;
```

```
    m[19]=5;
```

```
    /* virtual machine */
```

```
    pc=0;
```

```
    while(1) {
```

```
        printf(" pc=%4d reg=%4d\n", pc, reg);
```

```
        instr=m[pc]; pc=pc+1;
```

```
        switch(instr) {
```

```
            case 0: exit(0);
```

```
                break;
```

```
            case 1: printf("%d\n", reg);
```

```
                break;
```

```
            case 2: n=m[pc]; pc=pc+1;
```

```
                reg=n;
```

```
                break;
```

```
            case 3: add=m[pc]; pc=pc+1;
```

```
                reg=m[add];
```

```
                break;
```

```
            case 4: add=m[pc]; pc=pc+1;
```

```
                m[add]=reg;
```

```
                break;
```

```
            case 5: reg=reg+1;
```

```
                break;
```

```
            case 6: reg=reg-1;
```

```
                break;
```

```
            case 7: add=m[pc]; pc=pc+1;
```

```
                pc=add;
```

```
                break;
```

```
            case 8: add=m[pc]; pc=pc+1;
```

```

        if(reg==0){
            pc=add;
        }
        break;
    case 9: add=m[pc]; pc=pc+1;
        if(reg!=0){
            pc=add;
        }
        break;
    case 10: reg=pc;
        break;
    case 11: pc=reg;
        break;
    }
}
}
}

/*
2
*/

```

課題 3

```

#include <stdio.h>
#include <stdlib.h>

```

```

unsigned char m[256], pc, reg;

```

```

int main() {
    unsigned char instr,n, add;
    /* Machine Language */
    m[0]=3;
    m[1]=19;
    m[2]=8;
    m[3]=15;
    m[4]=6;
    m[5]=4;
    m[6]=19;
    m[7]=3;
    m[8]=18;
    m[9]=5;
    m[10]=1;
    m[11]=4;
    m[12]=18;
    m[13]=7;
    m[14]=0;
    m[15]=3;
    m[16]=18;
    m[17]=0;
    m[18]=0;
    m[19]=10;
    /* virtual machine */
    pc=0;
    while(1) {
        printf(" pc=%4d reg=%4d\n", pc, reg);
        instr=m[pc]; pc=pc+1;
        switch(instr) {
            case 0: exit(0);
                break;
            case 1: printf("%d\n", reg);

```

```

        break;
    case 2: n=m[pc]; pc=pc+1;
        reg=n;
        break;
    case 3: add=m[pc]; pc=pc+1;
        reg=m[add];
        break;
    case 4: add=m[pc]; pc=pc+1;
        m[add]=reg;
        break;
    case 5: reg=reg+1;
        break;
    case 6: reg=reg-1;
        break;
    case 7: add=m[pc]; pc=pc+1;
        pc=add;
        break;
    case 8: add=m[pc]; pc=pc+1;
        if(reg==0){
            pc=add;
        }
        break;
    case 9: add=m[pc]; pc=pc+1;
        if(reg!=0){
            pc=add;
        }
        break;
    case 10: reg=pc;
        break;
    case 11: pc=reg;
        break;
    }
}

/*
1
2
3
4
5
6
7
8
9
10
*/

```

課題4			No.	
			Date	
Loop: LDM Y	0:3	26		
JZB OVAR1	2:8	21		
STM X	4:4	25		
PCR	6:6			
Loop:2: LDM X	7:3	25		
JZB Loop	9:8	0		
PCR	11:6			
STM X	12:4	25		
LDM SUM	14:5	27		
JMP Loop2	16:5			
OVAR1: LDM SUM	14:4			
LDM SUM	16:5			
PR1	17:4	27		
HLT	19:7	7		
X: DC 10	21:3	27		
Y: DC 10	23:1			
SUM: DC 0	24:0			
	25:0			
	26:0			
	27:0			

課題4

```
#include <stdio.h>
#include <stdlib.h>
```

```
unsigned char m[256], pc, reg;
```

```
int main() {
    unsigned char instr, n, add;
    /* Machine Language */
    m[0]=3;
    m[1]=28;
    m[2]=8;
    m[3]=23;
    m[4]=6;
    m[5]=4;
    m[6]=28;
    m[7]=4;
    m[8]=27;
    m[9]=3;
    m[10]=27;
    m[11]=8;
```

```

m[12]=0;
m[13]=6;
m[14]=4;
m[15]=27;
m[16]=3;
m[17]=29;
m[18]=5;
m[19]=4;
m[20]=29;
m[21]=7;
m[22]=9;
m[23]=3;
m[24]=29;
m[25]=1;
m[26]=0;
m[27]=11;
m[28]=11;
m[29]=0;
/* virtual machine */
pc=0;
while(1) {
    printf(" pc=%4d reg=%4d\n", pc, reg);
    instr=m[pc]; pc=pc+1;
    switch(instr) {
        case 0: exit(0);
            break;
        case 1: printf("%d\n", reg);
            break;
        case 2: n=m[pc]; pc=pc+1;
            reg=n;
            break;
        case 3: add=m[pc]; pc=pc+1;
            reg=m[add];
            break;
        case 4: add=m[pc]; pc=pc+1;
            m[add]=reg;
            break;
        case 5: reg=reg+1;
            break;
        case 6: reg=reg-1;
            break;
        case 7: add=m[pc]; pc=pc+1;
            pc=add;
            break;
        case 8: add=m[pc]; pc=pc+1;
            if(reg==0){
                pc=add;
            }
            break;
        case 9: add=m[pc]; pc=pc+1;
            if(reg!=0){
                pc=add;
            }
            break;
        case 10: reg=pc;
            break;
        case 11: pc=reg;
            break;
    }
}

```


}

/*
55
*/

課題5			Date
Loop	LDM : Y	0:3	30
	JZE DWAR2	2:8	25
	DCR	4:6	
	STM Y	5:9	30
	LDM X	7:3	29
	STM Z	9:4	31
Loop2:	LDM Z	11:3	31
	JZE Loop	13:8	0
	DCR	15:6	
	STM Z	16:4	31
	LDM SUM	18:3	32
	ICR 3	20:5	
	STM SUM	21:4	32
	JMP Loop2	23:7	11
DWAR2:	LDM SUM	25:3	32
	PRT	27:1	
	HLT	28:0	
	X:7	29:7	
	Y:5	30:5	
	Z:0	31:0	
	SUM:0	32:0	

課題5

unsigned char m[256], pc, reg;

```
int main() {  
    unsigned char instr, n, add;  
    /* Machine Language */  
    m[0]=3;  
    m[1]=30;  
    m[2]=8;
```

```

m[3]=25;
m[4]=6;
m[5]=4;
m[6]=30;
m[7]=3;
m[8]=29;
m[9]=4;
m[10]=31;
m[11]=3;
m[12]=31;
m[13]=8;
m[14]=0;
m[15]=6;
m[16]=4;
m[17]=31;
m[18]=3;
m[19]=32;
m[20]=5;
m[21]=4;
m[22]=32;
m[23]=7;
m[24]=11;
m[25]=3;
m[26]=32;
m[27]=1;
m[28]=0;
m[29]=7;
m[30]=5;
m[31];
m[32]=0;
/* virtual machine */
pc=0;
while(1) {
    printf(" pc=%4d reg=%4d\n", pc, reg);
    instr=m[pc]; pc=pc+1;
    switch(instr) {
        case 0: exit(0);
            break;
        case 1: printf("%d\n", reg);
            break;
        case 2: n=m[pc]; pc=pc+1;
            reg=n;
            break;
        case 3: add=m[pc]; pc=pc+1;
            reg=m[add];
            break;
        case 4: add=m[pc]; pc=pc+1;
            m[add]=reg;
            break;
        case 5: reg=reg+1;
            break;
        case 6: reg=reg-1;
            break;
        case 7: add=m[pc]; pc=pc+1;
            pc=add;
            break;
        case 8: add=m[pc]; pc=pc+1;
            if(reg==0){
                pc=add;
            }
    }
}

```

```
        break;
    case 9: add=m[pc]; pc=pc+1;
        if(reg!=0){
            pc=add;
        }
        break;
    case 10: reg=pc;
        break;
    case 11: pc=reg;
        break;
    }
}

/*
35
*/
```

