ICS assignments 05

PB18030980

高海涵

Problem 1

a

Because the clock in display is slower than the clock running in the CPU so that during the time between two positive display clock, the content of the DSR register has been modified for many times. As a result, the content write into the DSR during this period will not be displayed

b

The display server program does not check for KBSR register for the device status

Problem 2

The character of the stack

First in, First out

Implementation of the stack

- hardware implementation, to maintain and modified the stack top pointer
- software implementation, to maintain an area of Memory and storage the top and base of the stack

Problem 3

a

Operation(push 1,pop 2)	letter
1	z
1	y
2	y
1	x
2	x
1	w
1	v
2	v
1	u
2	u
2	w
2	z
1	t
1	s
2	s
1	r
2	r
2	t

b

14 output stream

Problem 4

push and pop operation will operate 2 memory content in one movement,assume the stack is grown in opposite position

PUSH:

ADD R6, R6, #-2
STR R0, R6, #0
STR R1, R6, #1

POP:

LDR R0, R6, #0
LDR R1, R6, #1
ADD R6, R6, #2

Problem 5

a

u=z-(x*y+w)/v

b

`PUSH A PUSH B PUSH C SUB PUSH D ADD MUL PUSH A PUSH C ADD DIV POP E`

Problem 6

a

```
CLEAR:  ST  R2,TEMP
LEA R2,MASKS
ADD R2,R1,R2
LDR R2,R2,#0
NOT R2,R2
AND R0,R2,R0
LD  R2,TEMP
RET
TEMP:   .BLKW #1
```

b

```
CLEAR:  ST  R2,TEMP
LEA R2,MASKS
ADD R2,R1,R2
LDR R2,R2,#0
NOT R2,R2
NOT R0,R0
AND R0,R2,R0
NOT R0
LD  R2,TEMP
RET
TEMP:   .BLKW #1
```

Problem 7

```
        .ORIG x3000
        LD  R5, PTR
        LDI R6, CNT
        BRz DONEz
MORE    LDR R1,R5,#0
        ADD R5,R5,#1
        ADD R6,R6,#-1
        BRz DONE1
```

```
        LDR R2,R5,#0
        ADD R5,R5,#1
        ADD R6,R6,#-1
        BRz DONE2
        LDR R3,R5,#0
        ADD R5,R5,#1
        ADD R6,R6,#-1
        BRz DONE3
        LDR R4,R5,#0
        ADD R5,R5,#1
        ADD R6,R6,#-1
        BRnzp READY
DONEz   AND R0,R0,#0
        ADD R0,R0,#1
        BRnzp END

DONE1   AND R2,R2,#0
        ADD R2,R2,#1        ;R2 = 1
        ADD R3,R2,#0        ;R3 = 1
        ADD R4,R2,#0        ;R4 = 1
        BRnzp READY


DONE2   AND R3,R3,#0
        ADD R3,R3,#1        ;R3 = 1
        ADD R4,R4,#0        ;R4 = 1
        BRnzp READY

DONE3   AND R4,R4,#0
        ADD R4,R4,#1
READY   JSR mult_all
        ADD R6,R6,#0
        BRz END             ;checks CNT

        ADD R5,R5,#-1
        STR R0,R5,#0
        ADD R6,R6,#1
        BRnzp MORE
END     ST R0,RESULT
        HALT
RESULT        .BLKW 1
mult_all ...
             ...
             ...
             RET
PTR     .FILL x6001
CNT     .FILL x6000
.END
```

Problem 8

Doesn't save the return address when shift routine

Problem 9

FUN

Question 10

Marc 90
Jack 18

Mike 76

Question 11

a

```
SAVEREGISTERS ST R0, SAVER0
ST R3, SAVER3
ST R4, SAVER4
ST R5, SAVER5
ST R6, SAVER6
RET

 RESTOREREGISTERS LD R0, SAVER0

    LD R3, SAVER3
    LD R4, SAVER4
    LD R5, SAVER5
    LD R6, SAVER6
    RET
 SAVER0 .BLKW x1
 SAVER1 .BLKW x1
 SAVER2 .BLKW x1
 SAVER3 .BLKW x1
 SAVER4 .BLKW x1
 SAVER5 .BLKW x1
 SAVER6 .BLKW x1
```

b

calling program should save R7 for return address

Question 12

a

The full queue and empty can not be distinguished in our design

b

The largest capability of the queue should be n-1 or less

c

Question 13

The string BOBO is length 28. When the program stops,the value store in R3 is x0BE0