

CISP317 Practice Exam 2

Prof. Tak Auyeung

November 11, 2003

Instructions: You may bring any material that is handwritten or printed *prior* to the examination to help you. You can also bring a calculator if you think it may help you. However, you can only use the calculator for numerical computations only. You *cannot* let your calculator compile a program, or to communicate with others.

You, as an individual, are expected to do your own work. This means you cannot seek, receive or otherwise acquire any assistance except clarifications from the professor during an examination. Any communication involving the contents of the subject matter or the examination is considered cheating. Do not initiate or accept such communication, or the result of your examination is automatically voided.

New rules, read this! As of 2003.09.22, I no longer deduct points for wrong answers. Each correct answer is worth one point, each wrong answer is worth zero point, and each unanswered question is also worth zero point. This means you *should* guess and leave no question unanswered.

As a result, I also need to adjust the letter grade assignment break points. For your individual examination, “A” means at least 94%, “B” means at least 74%, “C” means at least 54%, “D” means at least 34% and “F” means below 34%. The break points for the final grade are now 26.83%, 48.5%, 70.17% and 91.83% as minimums for “D”, “C”, “B” and “A”, respectively.

Please note that this change does not affect your letter grade at all, it is just a number game to make some people feel better about guessing.

Make sure you write down you name on the upper right corner *first*, otherwise I cannot give points to anonymous students!

The baseline is XX, there are YY questions.

1 Assume the following byte sequence is at label `s1` and `s2` (bytes represented in hexadecimal digits):

```
s1: 61 62 63 64 00 66 67 68
s2: 31 32 33 00 35 36 37 38
```

What does the following code do?

```
ldi    r26,low(s1)
ldi    r27,high(s1)
ldi    r28,low(s2)
ldi    r29,high(s2)
ldi    r16,2

11:
ld     r17,X+
st     Y+,r17
cpi    r17,0
breq   l2
dec    r16
brne   l1
ldi    r17,0
st     Y+,r17
```

```
12:
nop
```

- A s1: 61 62 63 64 00 66 67 68
s2: 61 62 63 64 00 36 37 68
- B s1: 61 62 63 64 00 66 67 68
s2: 61 62 33 00 35 36 37 38
- C s1: 61 62 63 64 00 66 67 68
s2: 61 62 63 00 35 36 37 38
- D s1: 31 32 33 00 00 66 67 68
s2: 31 32 33 00 35 36 37 38
- E s1: 61 62 63 64 00 66 67 68
s2: 61 62 00 00 35 36 37 38

2 What is the result of the following instructions? Assume r0 has a value of 0x59 and r1 has a value of 0x31:

```

rol  r0
rol  r1

```

- A r0=0xb2, r1=0x62
- B r0=0xb?, r1=0x62, ? depends on the value of the C flag before the first instruction
- C r0=0xb2, r1=0x6?, ? depends on the value of the C flag before the first instruction
- D r0=0x24, r1=0x98
- E r0=0x?4, r1=0x98, ? depends on the value of the C flag before the first instruction

3 What is the minimum for ? to guarantee that r0 becomes zero at the nop instruction?

```

    ldi  r16,?
11:
    cpi  r16,0
    breq 12
    lsl  r0
    dec  r16
    rjmp 11
12:
    nop

```

- A 0
- B 7
- C 8
- D 255
- E 256

4 What is the value of r0 after these instruction? Assume r0=0xbd before these instructions:

```

lsr  r0
lsr  r0
lsr  r0
lsr  r0
lsl  r0
lsl  r0
lsl  r0
lsl  r0

```

- A 0xbd
- B 0x0b
- C 0xd0

D 0xb0

E cannot be determined because it depends on the initial value of the C flag

- 5 Assume `r16` contains the least significant byte of a number n , and `r17` contains the most significant byte of the same number. What is the value of `r18` after the following instructions?

```
    ldi    r20,3
11:   lsr    r17
      ror    r16
      ror    r18
      dec    r20
      brne  l1
      ldi    r20,5
12:   lsr    r18
      dec    r20
      brne  l2
```

A 3

B 5

C the dividend of $n \div 5$

D the remainder of $n \div 8$

E the dividend of $n \div 32$

- 6 What happens when the following code is executed? The `inc` instruction affects the N flag.

```
    sub    r0,r0
11:   inc    r0
      brlo  l1
      nop
```

A this code never gets to the `nop` instruction

B this code does not assemble

C when this code executes `nop`, `r0` has a value of `0x00`

D when this code executes `nop`, `r0` has a value of `0x80`

E when this code executes `nop`, `r0` has a value of `0xff`

- 7 As 8-bit signed numbers, which of the following represents the least value?

A 00001010_2

B 00100011_2

C 11001010_2

D 00001011_2

E 00110000_2

- 8 Which single instruction replaces the following instruction sequence without any change of behavior?

```
breq  l1
brne  l1
```

A `nop`

B `rjmp l1`

C `brcc l1`

D `brcs 11`

E `breq 11`

- 9 What is the 16-bit value stored in `r0` (LSB) and `r1` (MSB) after the following code executes? Assume the following byte sequence (in hexadecimal digits) is at label `m1`:

`m1: 01 00 00 10 02 00 00 20 00 01`

The code to analyze is as follows:

```
ldi r26,low(m1)
ldi r27,high(m1)
sub r0,r0
sub r1,r1
ldi r16,4
```

`l1:`

```
ld r2,X+
add r0,r2
ld r2,X+
adc r1,r2
dec r16
brne l1
```

(Reminder: LSB is the rightmost byte, MSB is the leftmost one.)

A `0x3021`

B `0x1202`

C `0x3013`

D `0x0100`

E `0x3103`

- 10 Which of the following instruction sequences end up at label `l1` if and only if `r16` has a value of 0? Assume the instruction following the sequences is not at label `l1`.

A `cpi r16,0`
`breq l1`

B `cp r16,r16`
`breq l1`

C `cpi r16,0`
`brne l1`

D `cp r16,r16`
`brne l1`

E `cpi r16,0`
`brcc l1`

- 11 Assume a block of memory is allocated as follows:

```
blk1: .byte 40
blk2:
```

What does the following code do?

```
ldi r26,low(blk1)
ldi r27,high(blk1)
ldi r16,low(blk2)
ldi r17,high(blk2)
sub r0,r0
```

```
11:
cp r26,r16
cpc r27,r17
brcc l2
st X+,r0
rjmp l1
```

```
12:
nop
```

- A initializes all 40 allocated bytes to zero
- B initializes all except the first of the 40 allocated bytes to zero
- C initializes all except the last of the 40 allocated bytes to zero
- D never terminates (infinite loop)
- E the code does not assemble

12 What are the values of flags after the following instruction, assuming $r0=0x54$ and $r1=0x5d$?

```
cp r0,r1
```

- A N=1, Z=0, C=0
- B N=1, Z=0, C=1
- C N=0, Z=1, C=0
- D N=0, Z=0, C=1
- E N=0, Z=1, C=1