Jose Franco Baquera

March 25, 2018

CS - 273

Homework #7

1. **Use logical and/or arithmetic rotation/shift instructions to divide the 32-bit *signed* number saved in R19, R18, R17, R16 (high byte first) by 2^7. We assume that R19 contains the most significant byte and R16 contains the least significant byte. Save the result in the same registers.**

Answer:

```
LDI R25, 7

LOOP: ASR R19

      ROR R18

      ROR R17

      ROR R16

      DEC R25

      CPI R25, 0

      BRNE LOOP

      RET
```

2. **Find the contents of R20 after each of the following is executed:**

```
LDI R20, 0x8A
CLC
ROR R20
ASR R20
```

- LDI R20, 0x8A; **The contents of R20 is 0x8A**, or 0b1000-1010.
- CLC; **The contents of R20 is still 0x8A**, or 0b1000-1010. C = 0.
- ROR R20; **The contents of R20 is 0x45**, or 0b0100-0101. C = 0.
- ASR R20; **The contents of R20 is 0x22**, or 0b0010-0010. C = 1.

The *final* contents of R20 at the end of the execution will be 0x22 or, in binary, 0b00100010.

3. **<u>Write a program to test if the pattern “1010” exists in an 8-bit input binary number. Please use shift instructions.</u>**

```
.data

.comm input, 1

.comm answer, 1

.global input, answer

.text

.global start

start:

   LDI R17, 8

   LDS R18, input // R18 contains the 8-bit input binary number.

stageZero:

   CPI R17, 0

   BREQ notFound

   DEC R17

   LSR R18

   BRCC stageOne

   RJMP stageZero

stageOne:

   CPI R17, 0

   BREQ notFound

   DEC R17

   LSR R18

   BRCC stageOne
```

```
    RJMP stageTwo
stageTwo:
    CPI R17, 0
    BREQ notFound
    DEC R17
    LSR R18
    BRCC stageThree
    RJMP stageZero
stageThree:
    CPI R17, 0
    BREQ notFound
    DEC R17
    LSR R18
    BRCC stageOne
    RJMP acceptingStage
acceptingStage: // This stage is reached if the pattern 1010 exists.
    LDI R19, 1 // 1 means true (i.e. the pattern was found).
    STS answer, R19
    RET
notFound: // This stage is reached if the pattern 1010 doesn’t exist.
    LDI R19, 0 // 0 means false (i.e. the pattern was not found).
    STS answer, R19
    RET
```

4. **Calculate the step sizes of ADC using the following number of bits with Vref=32V.**

   a. **2-bit:**

      Step size = 32 V / 2^2 = 32 V / 4 = **8 V.**

   b. **8-bit:**

      Step size = 32 V / 2^8 = 32 V / 256 = **0.125 V = 125 mV.**

   c. **16-bit:**

      Step size = 32 V / 2^{16} = 32 V / 65536 = **.00048828125 V = 0.48828125 mV.**

5. **With Vref=25.6V and 10 bit ADC, find Vin for the following outputs:**

NOTE: We can use the formula $D_{out} = \frac{V_{in}}{step\ size} \Rightarrow V_{in} = (step\ size) * D_{out}$
(The step size for this problem is 25.6 V/1024 = 0.025 V. Also, we use ***integer division***. Therefore, 5.375V and 5.38V will both produce Dout = 215, for example.)

   a. **Dout = 215**

      Lower V_{in} = (0.025V) * (215) = **5.375 V**

      Upper V_{in} = (0.025V) * (216) = 5.4 V. *We note that this upper bound is not included in the range. This is why we use a ')' instead of a ']' for the upper bound in the range.*

      Range Vin Answer: [5.375 V, 5.4 V)

   b. **Dout =84**

      Lower V_{in} = (0.025V) * (84) = **2.1 V**

      Upper V_{in} = (0.025V) * (85) = 2.125 V. *We note that this upper bound is not included in the range. This is why we use a ')' instead of a ']' for the upper bound in the range.*

      Range Vin Answer: [2.1 V, 2.125 V)

   c. **Dout = 233**

      Lower V_{in} = (0.025V) * (233) = **5.825 V**

      Upper V_{in} = (0.025V) * (234) = 5.85 V. *We note that this upper bound is not included in the range. This is why we use a ')' instead of a ']' for the upper bound in the range.*

      Range Vin Answer: [5.825 V, 5.85 V)

6. <u>For the 10-bit ADC in ATmega328p, find the step size for each of the following Vref</u> (*NOTE: The ADC in ATmega328p is 10 bits. In addition,* 2^{10} *= 1024):*

a. <u>25 mV</u>

Step Size = 25 mV/1024 = **0.0244140625 mV**

b. <u>4096 mV</u>

Step Size = 4096 mV/1024 = 0.004 V = **4 mV**

c. <u>50 V</u>

Step Size = 50V/1024 = **0.048828125 V = 48.828125 mV**