1. Convert the following to their equivalent 8-bit hexadecimal values:

- \(10111010\) \(\text{b}\) \(\rightarrow\) \(0xBA\)
- \(93\) \(\rightarrow\) \(0x5D\)
- \(-27\) (2’s complement format) \(\rightarrow\) \(0xE5\)

2. Give the range of numbers that can be represented in 10 bits using a 2’s complement number format.

\[-2^{(10-1)}\rightarrow +2^{(10-1)}-1 = -2^9\rightarrow +2^9-1 = -512 \rightarrow +511\]

3. Given a 40% duty cycle clock with a low clock width of 20 ns, what is the clock frequency in MHz?

If the clock has a 40% duty cycle, then 60% of the time the clock is low. Therefore 20ns/.6=33.33ns is the total clock period. The clock frequency is \(1/\text{period}=1/33.33\text{ns}=30\text{MHz}\).

4. Convert the following instructions into their machine language equivalent:

- \(\text{MOV.B} \ #0xE3,W4\) \(\rightarrow\) \(0xB3CE34\)
- \(\text{ADD} \ 0x0804,WREG\) \(\rightarrow\) \(0xB40804\)
- \(\text{MOV} \ [W2],W6\) \(\rightarrow\) \(0x780312\)

5. Assume the contents of data memory and working registers are as given below.

<table>
<thead>
<tr>
<th>Location</th>
<th>Value</th>
<th>Location</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>W0</td>
<td>0x1000</td>
<td>0x800</td>
<td>0xAA</td>
</tr>
<tr>
<td>W1</td>
<td>0x2000</td>
<td>0x801</td>
<td>0xBB</td>
</tr>
<tr>
<td>W2</td>
<td>0x3000</td>
<td>0x802</td>
<td>0xCC</td>
</tr>
<tr>
<td>W3</td>
<td>0x0803</td>
<td>0x803</td>
<td>0xDD</td>
</tr>
</tbody>
</table>

Give the value of the modified memory location or register for each of the following instructions:

- \(\text{MOV.B} \ [W3],W0\) \(\rightarrow\) \(0x10DD\)
- \(\text{MOV} \ W1,0x800\) \(\rightarrow\) \(0x2000\)
- \(\text{ADD} \ 0x0802,WREG\) \(\rightarrow\) \(0xEDCC\)
- \(\text{SUB} \ 0x0802,WREG\) \(\rightarrow\) \(0xCDCC\)

6. Write PIC24 assembly language equivalents for the following C code fragments:

```c
while (u16_i != u16_k) {
    loop body statements
}
```

```asm
loop_top:    mov   u16_k,WREG
             cp    u16_i
             bra   Z,loop_end
```
; loop body
bra   loop_top

loop_end:

while ((u8_x != 0) && (u8_y < u8_z)) {
  loop body statements
}

top_while:  clr.b WREG   ; W0=0
  cp.b  u8_x    ; u8_x=WREG
  bra  Z,end_while ; If u8_z=0 then end the while loop
  mov.b u8_z,W0  ; W0=u8_z
  cp.b  u8_y    ; u8_y=WREG
  bra GEU,end_while ; Skip while body if u8_y >= u8_z
  ; while body
  bra   top_while

end_while: