**B38DF Computer Architecture and Embedded Systems. Part II. Tutorial 2 – coding**

AVR Atmel Instruction Set (incomplete)

|  |  |
| --- | --- |
| Instruction | Description |
| LDI Rp, K | Load integer value K to register Rp |
| LDS Rp, k | Load contents of data at location k into Rp |
| CPI Rp, K  CP Rp, Rq | Compare the contents of register Rp with value K  Compare the contents of register Rp with the contents of Rq |
| ADD Rp, Rq | Adds the contents of Rq with Rp stores the result into Rp |
| SUB Rp, Rq | Subtracts the contents of Rq from Rp and stores the result into Rp |
| INC Rp | Increments the contents of Rp |
| DEC Rp | Decrements the contents of Rp |
| BRBS Z, k  BRBS N, k | Jump to location k if ALU operation results in a zero flag being set  Jump to location k if ALU operation results in a negative flag being set |
| BRGE k  BRLT k | Jump to location k if ALU operation results in Greater than or Equal to  Jump to location k if ALU operation results in Less than |
| BREQ k  BRNE k | Jump to location k if ALU operation results in Equal to …  Jump to location k if ALU operation results in Not Equal to … |
| RJMP k | Jump to location k |

|  |  |
| --- | --- |
| **sum=0,term=1;**  **for(i=1; i<=10; i++)**  **{**  **sum += term;**  **term += 1;**  **}** | **x=1;**  **while (x<16) do**  **{**  **// instructions to be repeated**  **x++;**  **}** |
| **.def sum = r16**  **.def term = r17**  **.def i = r18**  **ldi sum,0**  **ldi term,1**  **ldi i,1**  **again:**  **cpi i,11**  **brge exit**  **add sum,term**  **inc term**  **inc i**  **rjmp again**  **exit:**  **nop** | **LDI R18,1**  **L1:**  **CPI R18,16**  **BRGE L3**  **;instructions to be repeated**  **INC R18**  **RJMP L1**  **L3:** |

1. Translate the following C-code which calculates the sequence 0,1,2,5,12,29,70,… (Tn+1=2Tn+Tn-1) into AVR Atmel assembly code. Use instructions shown above.

**Tnm1=0, Tn=1, tmp=0;**

**for(i=1; i<8; i++)**

**{**

**tmp = Tn;**

**Tn = 2\*Tn + Tnm1;**

**Tnm1 = tmp;**

**}**

2. Develop an AVR Atmel assembly code to calculate S = 1+3+5+7+9+11+13. Use a loop.

3. Develop an AVR Atmel assembly code to implement

**if (R18 = 4 and R19 = 5), then R19 = 0. Otherwise R19 = 1.**

4. Using the HWU instruction set write a programme to execute the following formula

1. **A = ((B + C) AND D) - E** where D0 = B = 2, D1 = C = 5, D2 = D = 15 and D3 = E = 3.
2. Compile it as a series of hexadecimal numbers.

5. Using the HWU machine mnemonics write a programme to execute this programme fragment

**sum = 0xAA;**

**i = 0;**

**while (i < 20)**

**{**

**sum = sum ^ 0x20;**

**i= i+1;**

**}**

where D2=0x20 is hexadecimal 20 and D0=0xAA is hexadecimal AA and ^ is bit exclusive OR and D1=20 keeps the number of iterations.

6. Use the HW machine instruction set to code the following statements

a) **if (a < 0) sum = sum + 3; else sum = sum - 1;**

b) **case (a)**

**2: x = x + 2;**

**3: x = x - 3;**

**default x = 2;**

**endcase**

c) **y =10; x = 9; while (y > 0) { x = x – 1; y = y - 1; }**