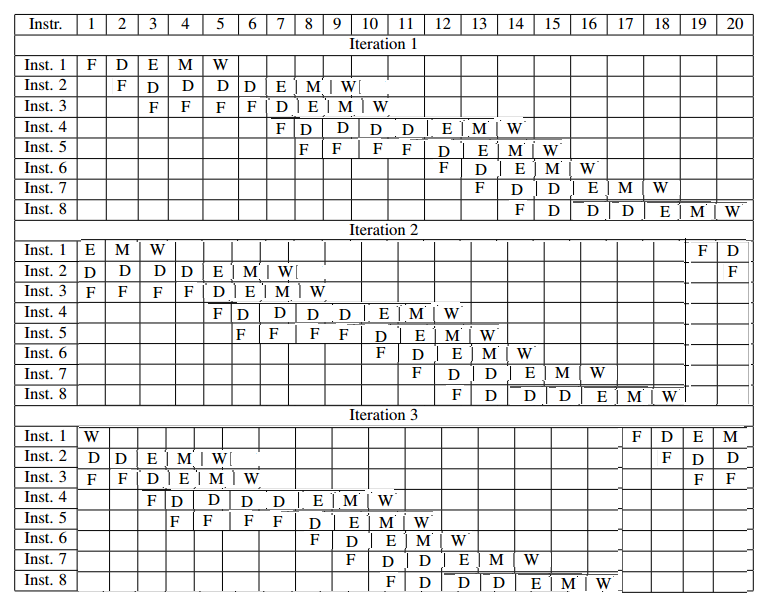
Problem 1:

|  |  |  |  |
| --- | --- | --- | --- |
|  | leave | iaddq | jmp |
| Fetch | icode:ifun <- M1[PC]=D:0  valP <- PC+1 | icode:ifun <- M1[PC]=C:0  rA:rB <- M1[PC+1]  valC <- M4[PC+2]  valP <- PC+6 | icode:ifun <- M1[PC]  valC <- M4[PC+1]  valP <- PC+9 |
| Decode | valA <- R[%rbp]  valB <- R[%rbp] | valB <- R[rB] | valA <- R[rA] |
| Execute | valE <- valB+4 | valE <- valC+valB  Set CC | valE <- valC + valA  Cnd <- Cond (CC, ifun) |
| Memory | valM <- M4[valA] |  |  |
| Write back | R[%rsp] <- valE  R[%rbp] <- valM | R[rB] <- valE |  |

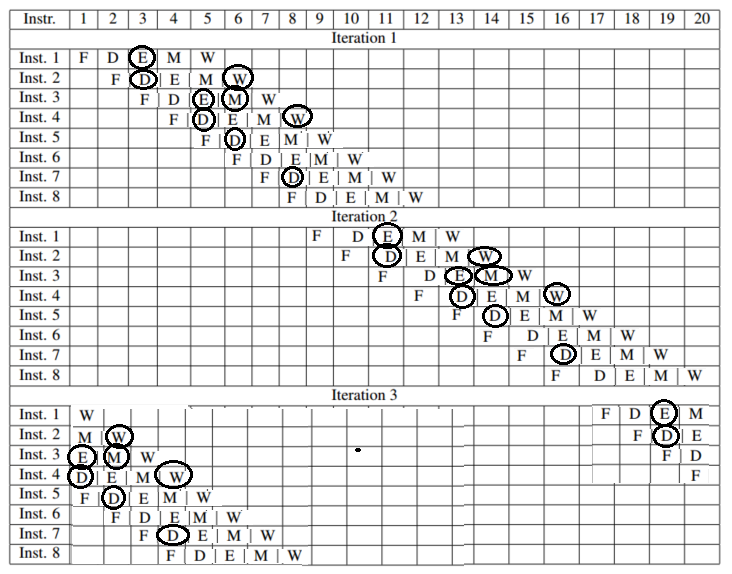
Problem 2:



CPI: 18 cycles / 8 instructions = 2.25 CPI

1 GHz : 1,000,000,000 cycles per second / 2.5 CPI = 44,444,444 instructions per second

Problem 3:



CPI : 8 cycles / 8 instructions = 1.0 CPI

2 GHz: 2,000,000,000 cycles per second / 1.0 CPI = 2,000,000,000 instructions per second

Problem 4:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CT | CT | CT | CT | CT | CT | CT | CT | CI | CI | CO | CO |
| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Address | Hit? | Read value |
| Read | 0x830 | Yes | 77 |
| Read | 0x408 | No |  |
| Read | 0xFFD | Yes | 97 |
| Read | 0x40B | No |  |

Problem 5:

Src array:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Col 0 | Col 1 | Col 2 | Col 3 |
| Row 0 | M | H | H | H |
| Row 1 | M | H | H | H |
| Row 2 | M | H | H | H |
| Row 3 | M | H | H | H |

Dst array:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Col 0 | Col 1 | Col 2 | Col 3 |
| Row 0 | M | M | M | M |
| Row 1 | M | M | M | M |
| Row 2 | M | M | M | M |
| Row 3 | M | M | M | M |

Problem 6:

Miss rates for N = 64:

SumA: 25

SumB: 100

SumC: 50