SOLUTIONS:

A.4

ALU instructions: \((30.2\% + 1.2\% + 1.2\% + 3.7\% + 6.8\% + 0.2\% + 0.4\% + 1.0\% + 1.6\%) = 46.3\%\)
Load-stores: \((16.0\% + 1.4\%) = 17.4\%\)
Conditional branches: 7.0%
Jumps: 0%
FP add: \((3.4\% + 1.4\%) = 4.8\%\)
Load-store FP: \((11.7\% + 4.4\%) = 16.1\%\)
Other FP: \((0.8\% + 0.4\% + 0.3\%) = 1.5\%\)

\[ \text{CPI} = 1.0 \times 46.3\% + 1.4 \times 17.4\% + 2.0 \times 7.0\% \times 60\% + 1.5 \times 7.0\% \times 40\% + 6.0 \times 6.4\% + 4.0 \times 4.8\% + 20 \times 0.4\% + 1.5 \times 16.1\% + 2.0 \times 1.5\% = 1.76 \]

A.7

```assembly
ex_a_7:  DADD   R1,R0,R0 ; R0 = 0, initialize i = 0
        SW    7000(R0),R1 ; store i
        loop:  LD     R1,7000(R0) ; get value of i
                DADDI  R2,R1,#3 ; R2 = word offset of B[i]
                DSLL   R2, R1, #3000 ; add base address of B to R2
                LD     R4,0(R3) ; load B[i]
                LD     R5,5000(R0) ; load C
                DADD   R6,R4,R5 ; B[i] + C
                LD     R1,7000(R0) ; get value of i
                DSLL   R2, R1, #3 ; R2 = word offset of A[i]
                DADDI  R7,R2,#1000 ; add base address of A to R2
                SD     0(R7),R6 ; A[i] ← B[i] + C
                LD     R1,7000(R0) ; get value of i
                DADDI  R1,R1,#1 ; increment i
                SD     7000(R0),R1 ; store i
                LD     R1,7000(R0) ; get value of i
                DADDI  R6,R1,#-101 ; is counter at 101?
                BNEZ   R6, loop ; if not 101, repeat
```

Total instructions executed are: \(2 + (16 \times 101)\) (since it is 101 iteration loop)

\[= 1618\]

There are 8 instruction related to memory = \(8 \times 101 = 808\)
C) & d) Considering 16 bit memory address and data operands.

So, both instruction and data sizes are 16 bits.

In accumulator, we have 1 load, 3 store and 8 instructions.

16 + 48 + 128 = 192 bits or 24 bytes data + code

In 64 bit,

64 + 192 + 512 = 768 bits or 96 bytes
In memory to memory,
We have 3 instructions and 9 memory operations
48 + 144 = 192 bits or 24 bytes
In 64 bit,
192 + 576 = 768 bits or 24 bytes
In stack,
We have 12 instructions and 9 memory operations
192 + 144 = 336 bits or 42 bytes
In 64 bit,
768 + 576 = 1344 bits or 168 bytes
In load/store,
We have 8 instructions and 5 memory operations
128 + 80 = 208 or 26 bytes
In 64 bit,
512 + 320 = 832 bits or 104 bytes.