Answers are evaluated based on correctness, completeness, and conciseness (i.e., answers must be correct, show how the result is obtained, and contain only the necessary text). If some specification is missing, just state your own specification and continue the problem according to your stated specification.

Problem 1

Answer the following questions.

Question 1.1

To be able to read the register values in the decode stage in parallel to decoding the instruction.

Question 1.2

Temporal locality: instructions in a loop, induction variables

Spatial locality: sequential instruction access (basic block), array data

Question 1.3

When a page is not in main memory and it has to be retrieved from the disk.

Question 1.4

A cache for fast translation from virtual to physical address.

Question 1.5

At AI=2.0, X2 is computation bound, while X4 is memory bound.

We can increase the memory bandwidth of X4.

Problem 2

Question 2.1

Yes, two: the register file and memory. For the register file: either split read and write into two parts within a single clock cycle (old school) or forward on a write when the same register is read within the register file. For memory: we use instruction and data caches to have one memory for fetch and one for the memory stage.

Question 2.2

ld x9, 64(x22) add x9, x21, x9 sd x9, 96(x22)

or 32-bit:

lw x9, 32(x22) add x9, x21, x9 sw x9, 48(x22)

Question 2.3

Solution is in the Solutions of 4.25, but here we use subi instead of addi in the solutions. And we use lw instead of ld.

Problem 3

Question 3.1

The first step is to convert from decimal to binary (sign-and-magnitude)

$$X = 1000|_{dec} = 11\ 1110\ 1000|_{bin} \quad \text{and} \quad Y = -993 = 11\ 1110\ 0001|_{bin}\ \text{(neg.)}$$

Since FP are normalized in $1.0 \leq X_{FP} < 2.0$, we obtain for the significand/magnitude of X and Y

$$M_X = 1.1111\ 0100\ 0|_{bin} \times 2^9 \quad \text{and} \quad M_Y = 1.1111\ 0000\ 1|_{bin} \times 2^9\ \text{(neg.)} \qquad (1)$$

binary32

To obtain the binary32 representation from (1) we have to add the bias to the exponent and omit the integer bit:

	sign	exp.	significand
X_{b32} :	0	1000 1000	1111 0100 0000 ... 000
Y_{b32} :	1	1000 1000	1111 0000 1000 ... 000

binary16

For binary16 the bias is 15.

	sign	exp.	significand
X_{b16} :	0	1 1000	1111 0100 00
Y_{b16} :	1	1 1000	1111 0000 10

Question 3.2

1) Significand alignment: $E_X - E_Y = 0 \Rightarrow$ significands are aligned.
2) Effective operation is subtraction:

M_X	1.1111 0100 0000 ... 000	-
M_Y	1.1111 0000 1000 ... 000	=
Sum	0.0000 0011 1000 ... 000	

3) Normalization is needed. Sum is shifted 7 position to the left and the exponent is decremented by 7. The sign is positive. No rounding needed.

	sign	exp.	significand
Z_{b32} :	0	1000 0001	1100 0000 0000 ... 000

Question 3.3

1) Add exponents:

$E_X + E_Y - Bias = E_Z \quad \Rightarrow \quad 24+24-15 = 33 > 30 = 1\,1110|_{bin}$ (max. exp. in binary16)

There is an overflow, and the sign is negative. The result is set to $-\infty$

	sign	exp.	significand
Z_{b16} :	1	1 1111	0000 0000 00

Problem 4

Question 4.1

Assuming the addresses given as byte addresses, each group of 16 accesses will map to the same 32-byte block so the cache will have a miss rate of 1/16. The miss rate is not sensitive to the size of the cache or the size of the working set. It is, however, sensitive to the access pattern and block size. All misses are compulsory misses.

Question 4.2

The miss rates are 1/8, 1/32, and 1/64, respectively. The workload is exploiting spatial locality.

Question 4.3

256 lines per way, 8 bit index, 4 bit offset, 20 bit tag.

8 KB size, 64 Kbit for data 20 x 512 = 10240 bits for tag memory

Problem 5

Question 5.1

We use the Amdahl's law:

$$\frac{t_{exe}(C0)}{t_{exe}(C0+C1)} = \frac{1}{F_S + \frac{F_P}{N}} \tag{2}$$

Since $F_P = 1 - F_S =$, by re-arranging and substituting the numerical values

$$\frac{75 \cdot 2}{100} = 1.5 = F_S + 1 \quad \Rightarrow \quad F_S = 0.5$$

Therefore, the parallel portion is $F_P = 1 - F_S = 50\%$.

Question 5.2

a) The denominator of (2) is

$$0.5 + \frac{0.5}{N=4} = 0.625 \quad \Rightarrow \quad \text{speed-up} = \frac{1}{0.625} = 1.6$$

b) By taking the limit $N \to \infty$ of (2)

$$\text{speed-up} = \lim_{N\to\infty} \frac{1}{F_S + \frac{F_P}{N}} = \frac{1}{F_S} = 2$$

Question 5.3

We apply again (2) with $F_S = 0.5/2 = 0.25$ and $F_P = 0.75$:

$$\frac{100}{x} = \frac{1}{0.25 + \frac{0.75}{2}} \quad \Rightarrow \quad x = 100 \cdot (0.25 + 0.375) = 62.5\ \mu s$$

Question 5.4

We call the two modes HP (high-performance) and LP (low-power), and we assume that the serial code is always executed on C0.

First, we determine how may cycles are spent in the serial part:

$$\text{Total cycles:} \quad (100 \times 10^{-6}) \cdot (2 \times 10^9) = 200,000 \text{ cycles.}$$

Serial part 50,000, parallel part 150,000 cycles (divided on two cores). The total cycles executed on C0 are 50,000+150,000/2=125,000.

The execution time is the maximum between t_{C0} an t_{C1}

C0	C1	t_{C0}	t_{C1}	$\max(t_{C0}, t_{C1})$
HP	HP	$125 \cdot 0.5 = 62.5$	$75 \cdot 0.5 = 37,5$	62.5 μs
HP	LP	$125 \cdot 0.5 = 62.5$	$75 \cdot 1.0 = 75,0$	75.0 μs
LP	HP	$125 \cdot 1.0 = 125$	$75 \cdot 0.5 = 37,5$	125.0 μs
LP	HP	$125 \cdot 1.0 = 125$	$75 \cdot 1.0 = 75.0$	125.0 μs

 ... continued on the next page

Question 5.5

The energy consumption to run A2 is

$$E_{A2} = t_{C0} \cdot P_{C0} + t_{C0} \cdot P_{C0} \quad [J]$$

C0	C1	E_{A2}		
HP	HP	$62.5 \cdot 10 + 37,5 \cdot 10$	=	1.000 mJ
HP	LP	$62.5 \cdot 10 + 75.0 \cdot 4$	=	0.925 mJ
LP	HP	$125 \cdot 4 + 37,5 \cdot 10$	=	0.875 mJ
LP	LP	$125 \cdot 4 + 75.0 \cdot 4$	=	0.800 mJ

Therefore, the lowest energy consumption is 0.8 mJ.

End of the Exam