Norwegian University of Science and Technology
Department of Computer and Information Science

Contact:
Magnus Jahre (952 22 309)

TDT4260/DT8803 COMPUTER ARCHITECTURE EXAM

Monday 4. June
Time: 09:00 – 13:00
ENGLISH

Allowed Aids:
D.
No written or handwritten examination support materials are permitted.
A specified, simple calculator is permitted.

Use the provided space to answer the problems. If you need more space, an extra answer box is available on the last page of the exam. The exam accounts for 80% of the final grade, and the provided points show the maximal number of points that can be achieved on each assignment. Read the problem texts throughly. You can answer the questions in English or Norwegian.

Candidate Number:

Problem 1 Multiple Choice (30 points)

Answer by circling the answer alternative you believe is the correct answer. You are awarded 3 points for a correct answer and 0 points if you do not answer. If your answer is wrong or you circle more than one alternative, you will get -1 point.

a) (3 p) Which of the following trends are *not* correct?

1. The number of transistors on a chip increases by between 40% and 55% per year
2. Single processor performance improved by 52% per year between 1986 and 2000
3. Single processor performance is currently doubling every two years
4. Dynamic Random Access Memory (DRAM) latency improves by 7% per year

Riktig svar: Alternativ 3

b) (3 p) Which of the following performance improvement techniques does *not* exploit parallelism?

1. Increasing the clock frequency by increasing the pipeline depth
2. Reducing the cache access latency by making the cache smaller
3. Increasing instruction throughput by adding more functional units
4. Improving system performance by adding an additional processor core

Riktig svar: Alternativ 2

c) (3 p) Which of the following categories is *not* part of Flynn's taxonomy?

1. SISD
2. SIMD
3. SIMT
4. MISD

Riktig svar: Alternativ 3

Candidate Number:

d) (3 p) Assume that 5% of the runtime of a certain application cannot be parallelized. What is the maximum possible speedup by parallelization for this application if we assume that the input set is fixed?

1. 2
2. 10
3. 20
4. 100

Riktig svar: Alternativ 3

Solution: Amdahl's law gives $\frac{1}{s+\frac{p}{n}}$, and when $n \to \infty$ we get $\frac{1}{s} = \frac{1}{0.05} = 20$

e) (3 p) Assume the same application as in Assignment 1 d). What happens to the speedup if we scale the data set and keep execution time fixed?

1. The speedup is significantly higher than with a fixed data set
2. The speedup is roughly the same as with a fixed data set
3. The speedup is significantly lower than with a fixed data set
4. Scaling the data set is impossible for most practical parallel applications

Riktig svar: Alternativ 1

Solution: This question can be answered with Gustafson's law: $S = n + (1 - n)s'$ which gives a significantly better speedup than with Amdahl's law. For instance, the Amdahl speedup with $n = 20$ is $\frac{1}{0.05+\frac{0.95}{20}} = 10.3$ and the Gustafson speedup is $S = 20 + (1 - 20)0.05 = 19.05$. The difference increases with n.

f) (3 p) Which of the following ratios is *not* part of the CPU performance equation (also known as the "Iron Law")?

1. Instructions/Program
2. Program/Seconds
3. Clock Cycles/Instruction
4. Seconds/Clock Cycle

Riktig svar: Alternativ 2

Candidate Number:

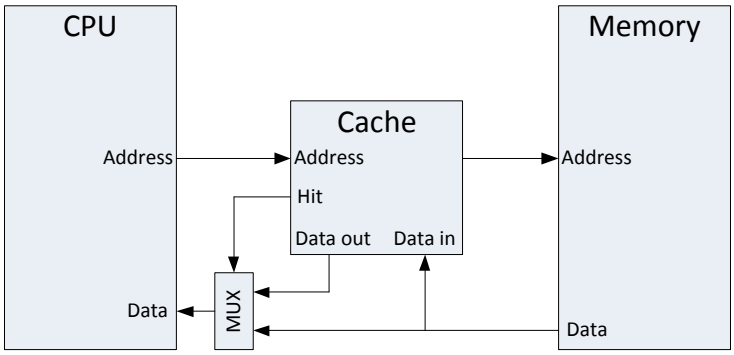

Figure 1: Cache System Block Diagram A

g) (3 p) Which of the following statements are *not* currently considered to be a challenge for future multi-core architectures?

1. The power budget of the chip is fixed and will remain so in the future
2. Architects need to control memory hierarchy power consumption to meet the power budget
3. Energy efficient parallel software must be developed
4. Architects need to find new aggressive ILP techniques to improve single thread performance

Riktig svar: Alternativ 4

h) (3 p) Consider the system outlined in Figure 1. Assume a 1 clock cycle cache latency, a 20 clock cycle memory latency and that the cache access must be completed before memory is accessed. What is the average access time for an application with a 90% cache hit rate?

1. 1.0
2. 2.9
3. 3.0
4. 20.0

Riktig svar: Alternativ 3

Solution: Cache access and memory access is serialized. Consequently, all accesses have the 1 cycle cache access latency and 10% of the accesses have the 20 cycle memory latency. Therefore: $1 + 0.1 \cdot 20 = 1 + 2 = 3.0$

Candidate Number:

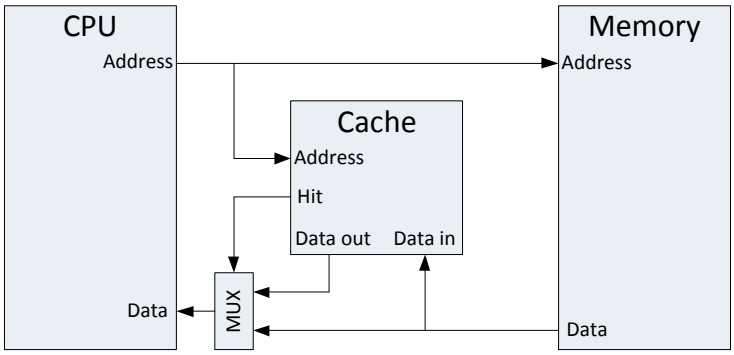

Figure 2: Cache System Block Diagram B

i) (3 p) Consider the system outlined in Figure 2. Assume a 1 clock cycle cache latency and a 20 clock cycle memory latency. What is the best average access time for an application with a 90% cache hit rate?

1. 1.0
2. 2.9
3. 3.0
4. 20.0

Riktig svar: Alternativ 2

Solution: Cache access and memory access is in parallel. Consequently, 90% of the accesses have the 1 cycle cache access latency and 10% of the accesses have the 20 cycle memory latency. Therefore: $0.9 \cdot 1 + 0.1 \cdot 20 = 0.9 + 2.0 = 2.9$

j) (3 p) Which statement about cache coherency protocols is *not* correct?

1. Write invalidate protocols are much more common that write update protocols.
2. Write update protocols consume more bandwidth that write invalidate protocols.
3. Snooping cache coherency protocols use a broadcast interconnect as the point of serialization
4. It is impossible to implement a directory-based cache coherency protocol in a system with a bus interconnect

Riktig svar: Alternativ 4

Candidate Number:

Problem 2 Memory Systems (10 points)

a) (5 p) Draw a block diagram of a 2-way set associative cache. Explain any necessary assumptions.

Solution: See textbook page B-13 or slide 26, lecture 2b

b) (5 p) Draw a block diagram of a system that implements the virtually indexed/physically tagged optimization. Explain any necessary assumptions.

Solution: See textbook B-39 or slide 39, lecture 2b

```
1 LW    R1, 0(R2)
2 SUB   R4, R1, R5
3 AND   R1, R4, R5
```

Figure 3: Assembly Code 1

Problem 3 Instruction Level Parallelism (10 points)

a) (5 p) Find and name all dependencies in Figure 3. The first register in the operand list is output for all instructions.

Solution: True data dependency: Line 2 to 1, R1.
True data dependency: Line 3 to 2, R4.
Anti-dependency: Line 3 to 2, R1.
Output-dependency: Line 3 to 1, R1.

```
1 LW      R6, 32(R1)
2 LW      R2, 36(R1)
3 MUL     R0, R2, R4
4 SUB     R8, R2, R6
5 DIV     R10,R0, R6
6 ADD     R7, R8, R2
```

Figure 4: Assembly Code 2

Format 1:	LW/SW	LW/SW	ADD/SUB	ADD/SUB
Format 2:	MUL/DIV	MUL/DIV	ADD/SUB	ADD/SUB

Table 1: VLIW Instruction Format

b) (5 p) A VLIW instruction format is given in Table 1. Each slot can be filled with an instruction of the given type or a no-operation (NOP). Find a schedule of the code in Figure 4 that minimizes execution time. State any necessary assumptions.

Solution: A minimal schedule is:

Format 1:	LW 1	LW 2	NOP	NOP
Format 2:	MUL 3	NOP	SUB 4	NOP
Format 2:	DIV 5	NOP	ADD 6	NOP

Add 3 and Sub 4 both depend on LW 1 and LW2, and must be placed in subsequent instructions. There is no format that combines MULTs and DIVs with load/stores, but regardless they both depend on LW1 or LW2. Therefore, these are placed in subsequent instructions. Then, Div 5 depends on MUL 3 and Add 6 depends on SUB 4 which result in these being placed in the last instruction.

Candidate Number:

```
1 LV       V1, Rx       ;load vector x
2 MULVS.D  V2, V1, F0   ;vector-scalar multiply
3 LV       V3, Ry       ;load vector y
4 ADDVV.D  V4, V2, V3   ;add x and y
5 SV       V4, Ry       ;store the sum
```

Figure 5: Vector Assembly Code for DAXPY

Problem 4 Data-Level Parallelism (10 points)

Assume a vector computer with 1 vector load/store unit, 1 vector add/subtract unit, 1 vector multiply/divide unit and a vector length of 64. The vector computer implements *chaining*, and accessing off-chip memory takes 20 clock cycles. You can ignore effects resulting from start-up latencies in this assignment.

a) (5 p) What is lowest execution time in clock cycles of the code Figure 5? Make any necessary assumptions.

Solution: Since we only have one load/store unit, the LVs cannot be executed concurrently. Chaining makes it possible to execute dependent instructions concurrently.

This gives the following schedule:
1. LV 1, MULVS.D
2. LV 3, ADDVV.D
3. SV

Since we can ignore startup overhead, each of these instructions take 64 clock cycles (i.e. the vector length). Consequently, the total execution time is $64 \cdot 3 = 192$ clock cycles.

b) (5 p) How many memory banks are needed to sustain maximum performance for this machine? Explain your reasoning.

Solution: To keep the computer operating at full capacity, the load store unit needs to deliver a new data element every clock cycle. This can be achieved by pipelining, but to accomplish this we need enough independently accessable memories (i.e. banks) to completely hide the memory latency. This is one bank for each clock cycle of latency, and for this computer, we need at least 20 memory banks.

Candidate Number:

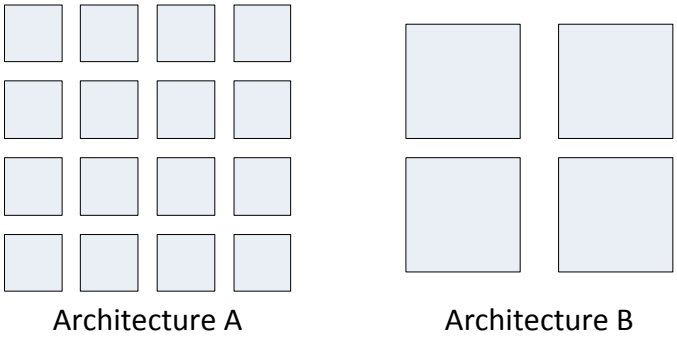

Figure 6: Multi-core Processors

Problem 5 Thread-Level Parallelism (10 points)

Figure 6 shows two multi-core processor architectures (Architecture A and Architecture B) that have the same area and power budget. For simplicity, assume that the performance of each of the small cores in Architecture A is 1 performance unit and that it consumes 1 power unit. A processor with unit 1 performance is able to execute 1 work unit every time unit.

a) (5 p) What is the expected performance of a processor core in Architecture B? Make any necessary assumptions and explain your reasoning.

Solution: To answer this question, we can use Pollack's rule which states that the performance of processor cores increase with the square root of the area. The cores in Architecture B are 4 times larger than in Architecture A, and the square root of 4 is 2. Consequently, Architecture B cores have a performance of 2 performance units.

b) (5 p) Assume a scalable parallel program with 32 work units. Which architecture is most suitable for this program in terms of performance and power efficiency? Explain you reasoning and make any necessary assumptions.

Solution: Architecture A execution time: $(32/16)/1 = 2$ time units
Architecture B execution time: $(32/4)/2 = 8$ time units

Architecture A has higher performance for this program. Since the power budget for both architectures is the same, it is also more power efficient.

Candidate Number:

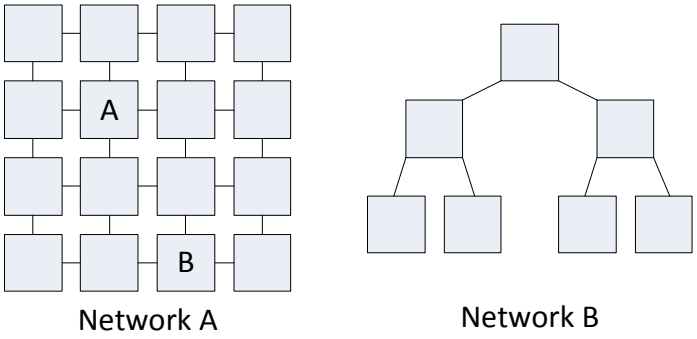

Figure 7: Interconnection Network Examples

Problem 6 Interconnection Networks (10 points)

a) (5 p) Assume Network A in Figure 7, the information in Table 2 (on page 11) and a message size of 1024 bytes. What is the lowest transfer latency in clock cycles between points A and B? Make any necessary assumptions, and explain your reasoning. Control and status information is transmitted over separate lines and can be ignored in this assignment.

Solution: There are two shortest paths from A to B, and both traverse 3 links and two intermediate routers. We assume that we need to traverse the routers at both end points which gives a sender overhead and receiver overhead of 1 cycle, respectively. The 1024 byte payload is divided into $\frac{1024}{64} = 16$ packets. Thus, transmitting the first packet takes 7 cycles. Then, a new data element will arrive each cycle after that for 15 additional cycles. Thus, the total transfer latency is 22 clock cycles.

Data	Value
Link latency	1 cycle
Router latency	1 cycle
Link data width	64 byte
Clock Frequency	2 GHz

Table 2: Network Properties

b) (5 p) Find the bisection bandwidth of networks A and B in Figure 7 using the information in Table 2. Make any necessary assumptions.

Solution: Bisection bandwidth is the minimum aggregate bandwidth that crosses a cut that partitions the network into two groups of roughly the same size.

Candidate Number:

A clock frequency of 2 GHz is $2 * 10^9$ Hz which gives a cycle time of $1/2 * 10^9 = 0.5 * 10^{-9}$ seconds. The link bandwidth is the number of bytes transferred each second which gives $64/(0.5 * 10^{-9}) = 128 * 10^9$ bytes per second or 128 GB/s.

In Network A there are many minimal cuts, but all cross 4 links. Consequently, the bisection bandwidth of network A is $4 \cdot 128 = 512$ GB/s.

In Network B, there are two minimal cuts where each cuts one of the links from the root node. Consequently, the bisection is one link wide and the bisection bandwidth is 128 GB/s.

Additional Answer Space

Candidate Number: