# Quiz 2

## CS 3853

**August 07, 2024**

**Name (please print):** _____

**Student ID:** _____

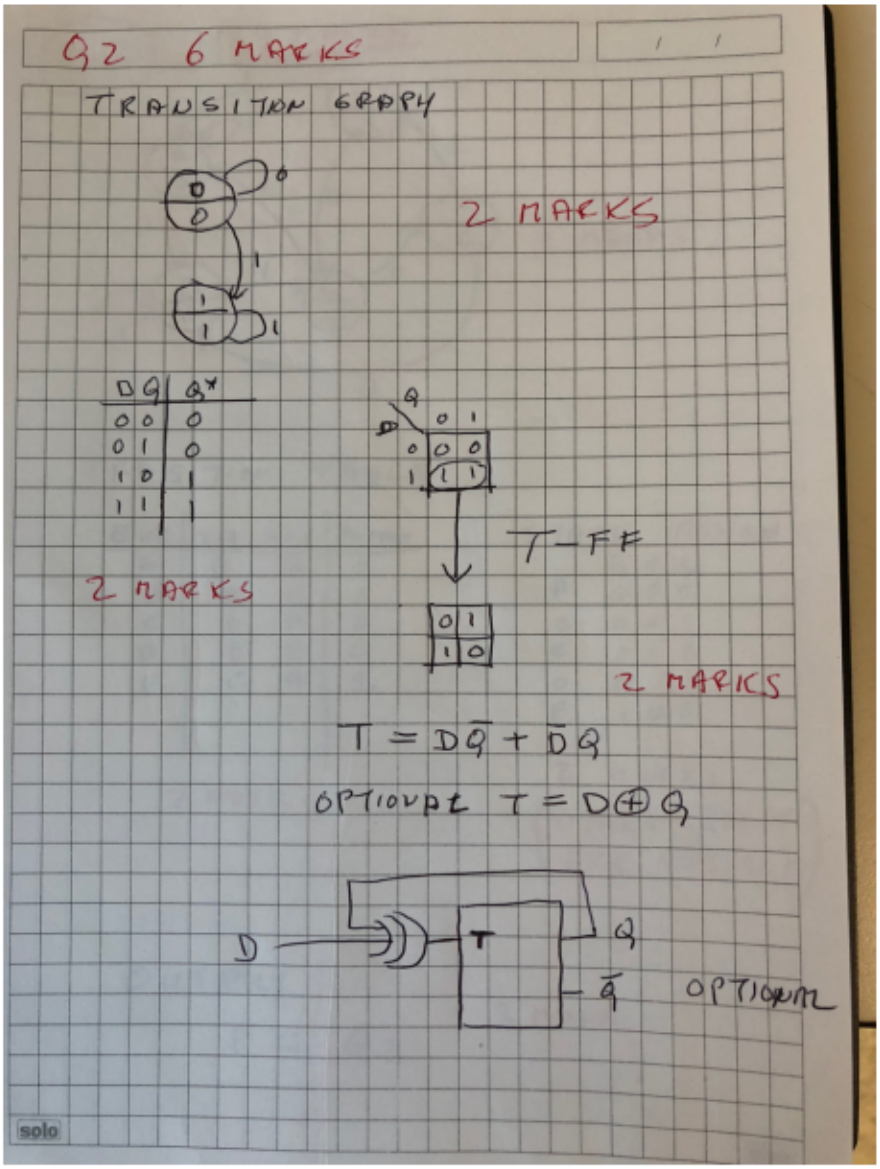
**Total points: 30**

**Duration: 1 hour**

**Instructions:**

- Please answer all questions
- The quiz is open book, and you can use the internet
- Individual work is expected, therefore group discussion is prohibited
- You can check your lecture notes, your own notes and the text book during the quiz
- You can use calculator as well the IAS instruction set

**1.** (*6 points*) Implement a D-FF using a T-FF and any additional gates that you may need. Follow the formal procedures described in class.




Give full marks even without the XOR solution. No circuit needs to be drawn.

**2.** (*4 points*) We have 16 KiB of RAM and two options for our cache. Cache A is size 1 KiB, with 256 B blocks and is direct-mapped. Cache B is size 2 KiB, also with 256 B blocks and is 4-way set associative.

- Show the address format for cache A (2 points).
  **Solution: 0.5 points for each of the three computations, tag, index and offset. 0.5 for showing the address format**
  Offset bits = $log_2 256 = 8$
  Index = $2^{10}/2^8 = 2$
  For 14 address bits
  Tag = 14 - 2 - 8 = 4
  Note: If anyone assumes any other number of bits, will be graded accordingly

- Show the address format for cache B (2 points).
  **Solution: 0.5 points for each of the three computations, tag, index and offset. 0.5 for showing the address format**
  Offset bits = $log_2 256 = 8$
  Blocks = 2 KiB = 8
  sets = 4 blocks/set = 1 set
  For 14 address bits
  Tag = 14 - 8 - 1 = 5
  Note: If anyone assumes any other number of bits, will be graded accordingly

**3.** (*4 points*) Given the following instruction sequence:

```
I1:  R1 = 100
I2:  R1 = R2 + R5
I3:  R2 = R4 - 25
I4:  R4 = R1 + R3
I5:  R1 = R1 + 30
```

Identify the:

- Write-After-Read (WAR) dependencies (2 points).
- Write-After-Write (WAW) dependencies (2 points).

**4 points - 1 for each dependency**

**WAW - write after write** - {I1, I2}

**WAR - write after read** - {I2, I3}, {I3, I4}, {I4, I5},

**4.** (*6 points*) Is it possible to design an expanding opcode to allow the following to be encoded in a 32 bit instruction? A register is 8 bits.
**6 points - 1 for computing bits for each instruction, the total bits and the specified instruction bits (32) points, then 1 for the conclusion**

- 4 instructions with 3 registers.
  $4 \times 2^8 \times 2^8 \times 2^8 = 4 \times 2^{24} = 2^{26} = 67108864$ bit patterns
- 255 instructions with one register.
  $255 \times 2^8 = 65280$ bit patterns
- 16 instructions with zero registers.
  16 bit patterns

Total bits = 67108864 + 65280 + 16 = 67174160
With 32 bit instruction we can have $2^{32} = 4294967296$ bit patterns
Required bit patterns (67174160) is less than what we have (4294967296), so this instruction set is possible with 32 bits

**5.** (*10 points*) A computer has a single $L_1$ cache with a 2 ns hit time and a 2% miss rate. Main memory has a 60 ns access time. If we add an $L_2$ cache with a 1.5 ns hit time and a 6% miss rate.

- What is the computer's effective access time without the $L_2$ cache (3 points).
  $AMAT = Hit\ Time + Miss\ Rate \times Miss\ Penalty$
  $AMAT = T_{hit}(L1) + Miss\%(L1) \times T(Mem)$
  = 2 ns + .02 * 60 ns = 3.2 ns

- What is the computer's effective access time with the $L_2$ cache (4 points).
  $AMAT = Hit\ Time_{L1} + Miss\ Rate_{L1} \times (Hit\ Time_{L2} + Miss\ Rate_{L2} \times Miss\ Penalty_{L2})$
  = 2 ns + .02(1.5ns + 0.06(60ns)) = 2.1 ns

- How much of a speedup does the $L_2$ cache give the computer (3 points).
  = 3.2/2.1 = 1.52
  The $L_2$ speeds up the computer by 1.5×