EDA322/ DIT797: Digital Design
Exam - March 2025

Date: March 19, 2024

Time: **14:00-18:00**

Examiner: Ioannis Sourdis (substituted during the exam by Ahsen Ejaz)

Department: Computer Science and Engineering

Inquiries: contact through phone Ahsen Ejaz, phone extension 5232

Duration: 4 hours

Grading scale: 100 points in total

U: 0%-49%,
3: 50%-64%,
4: 65%-84%,
5: 85%-100%

Available references: an A4 paper sheet (2 pages) with student notes, a calculator, are allowed.

General: Submit your solutions, in English, on blank paper sheets. Write legibly; feel free to use figures to get your point across.

The order of answering the questions does not matter (start with the easiest ones).

Please start the solutions for each problem on a new sheet. Please number the sheets so that the solutions are in numerical order.

Note that it is possible to receive partial credit for an answer even if it is not 100% correct.

Your personal identity code is required on each submitted sheet!

Good luck!

Note: *These are example answers to the exam questions for one possible set of randomized variables.*

Question 1 Arithmetic: (10 points)

- Consider that the delay of a full-adder is equal to the delay of a 2-to-1 multiplexer. Find the block sizes (i.e. the number of bits per block) that minimize the delay of a 32-bit carry-select adder (consider variable-size blocks).
- How does that delay compare to the delay of a 32-bit ripple carry adder?
- Consider that the area of a full adder is double the area of a 2-to-1 multiplexer. How does the area of your carry select adder compare with the area of a ripple carry adder?

Answer

If FA delay = MUX delay = x then

The best partitioning of a carry select adder would be [8, 7, 5, 4, 3, 2, 2] (more answers may lead to minimal delay, e.g. [7, 7, 6, 5, 4, 2, 1] or [6, 5, 4, 4, 4, 4, 2, 2]))

Which requires 8 FA delays plus one MUX delay = 9*x in total

A ripple carry adder would need 32*x, so the carry select would be 9/32 the delay of a ripple carry.

Let FA area = MUX area = y then:

A z-bit block of the carry select adder would need z FAs and z+1 muxes so (2*z+1)y area, except the very first block which needs only z FAs so it has area z*y.

So the area of the 32-bit carry select is =

8*2y +y + 7*2y+y +5*2y+y +4*2y+y + 3*2y+y + 2*2y+y +2y = 30*2*y + 8y = 68*y

The area of a ripple carry adder would be 32*y, so the carry select would have 68/32 the area of a ripple carry.

Question 2 Reconfigurable Computing: (10 points)

Draw the block diagram of a 4-input FPGA logic cell. If all configuration bits are stored in SRAM cells, how many SRAM cells are needed in total?

Answer

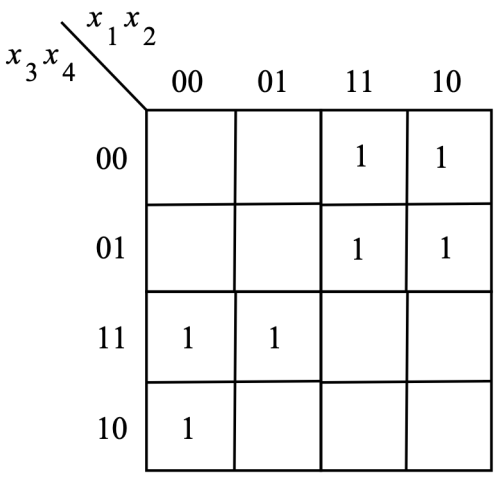
16 for the LUT,

1 for the FF multiplexer

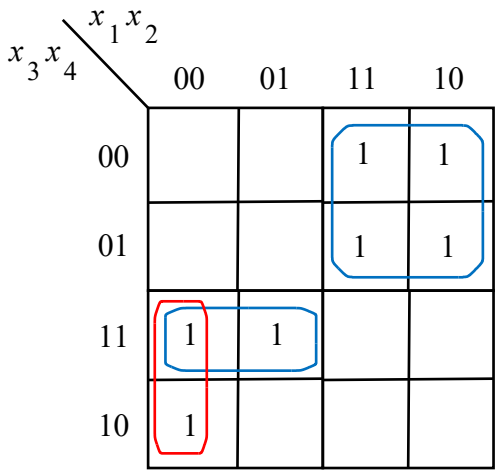
Total 17.

Question 3 Hazards: (10 points)

- Minimize the function in the following K-map
- Find and remove all hazards.



Answer



Question 4 Number representation: (10 points)
Suppose you need to represent distance from 1 to 7 meters (m) with a maximum relative error of 2%. Find a number representation with minimum number of bits that satisfies the above error constraint.

Answer
At the low end of the interval, 2% of 1 m is 2 cm,
At the other end of the interval, 2% of 7m is 14 cm,

a) If we chose a fixed-point representation then: 2 cm absolute error needs a resolution of 4 cm = 1/25 m.

If meters is the integer part we need, then we need 3 bits for integer (to count between 1-7 meters) because $2^3 = 8$. Then, the fraction needs 5 bits to count $1/2^5 = 1/32$ meters (which is a bit shorter than the 1/25 meters resolution). So the fixed-point representation would require 8 bits (3.5).

Alternatively, we can count multiples of 4 cm which is the required resolution. 7 meters = 175 * 4cm for which we need again 8 bits (2^8=256).

b) For a floating-point representation, the mantissa needs to be only 5 bits to offer accuracy of 2% = 1/50 because the resolution needs be double that: 1/25; $1/32 = 1/2^5$ should then be enough (5 bits mantissa). The dynamic range is 7m which is smaller than 2^3=8. So, 2 bits exponent is enough to represent 2^3. We can finally set the bias to b=0 because we need to use the maximum value of the exponent. So the floating point number needs in total 5+2 = 7 bits, one less than the fixed point.

Question 5 Pipelining and Adders: (10 points)
Consider an unpipelined 8-bit Ripple Carry Adder. Consider also that the delay of a full adder (FA) is 1.8 ns. In addition, the setup time of a flip-flop is 100 ps and the propagation time is 100ps.

a) What is the latency, maximum operating frequency, and throughput of the unpipelined version? Consider that inputs and outputs of the unpipelined version are registered.
b) Pipeline the adder to maximize throughput without increasing latency more than 10%. What is the maximum throughput and what is the latency and maximum operating frequency of the pipelined adder?

Answer

a) latency of unpipelined RCA: 8*1.8 + 0.1 +0.1 = 14,6 ns

max operating frequency: 1/14,6 GHz ~= 0.0685 GHz = 68.5MHz

Throughput is one addition per 14,6ns = 68.5Mops/sec

b) 10% of the above latency is 1.46 ns, which allows to have a maximum total latency of 16.06 ns.

To find the number of stages (n) that maximizes the throughput,
n ((8/n)*1.8 + 0.1 + 0.1) < 16.06, which solves for n < 8.3.

This means that each stage will fit one FA achieving throughput of one result every 1.8ns+100ps+100ps = 2 ns, which is 1/2GHz = 500 MHz

The latency of the pipelined design would be 8 times the latency of a stage 8*(1.8+0.1+0.1) ns= 16 ns < 14.6* 110% = 16.06 ns

Question 6 FSMs: (10 points)
Design a Moore FSM for a vending machine that receives coins of type-1 (2 SEK) and type-2 (1 SEK) and opens when it receives the total amount of 5 SEK. If it receives a larger amount it resets and gives back all the coins it has collected so far.

The FSM should use D flip flops with a "**reset"** and a "**load-enable"** input.

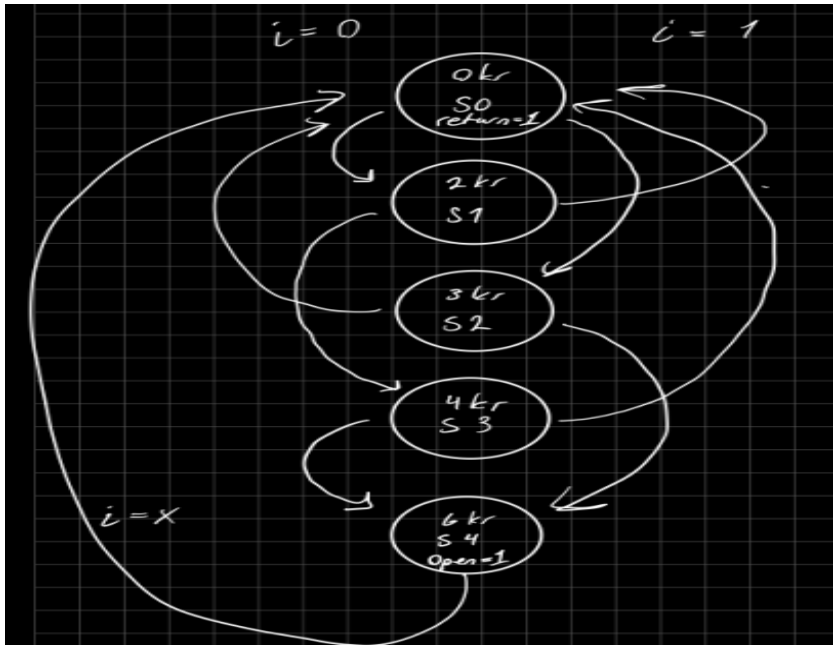
The FSM uses an 1-bit input called "**coin-type"**, which is "0" when a type-1 coin is inserted and "1" when a type-2 coin is inserted. When the correct amount is received, the machine reaches its final state, where the output "**open"** is set to "1" in order to accept the received coins and give away the product. After this final state the machine goes back to its initial state. When a larger amount than the expected one is received, then the FSM returns to its initial state. At the initial state a second output called "**return"** is set to "1", which returns all coins received during the transaction.

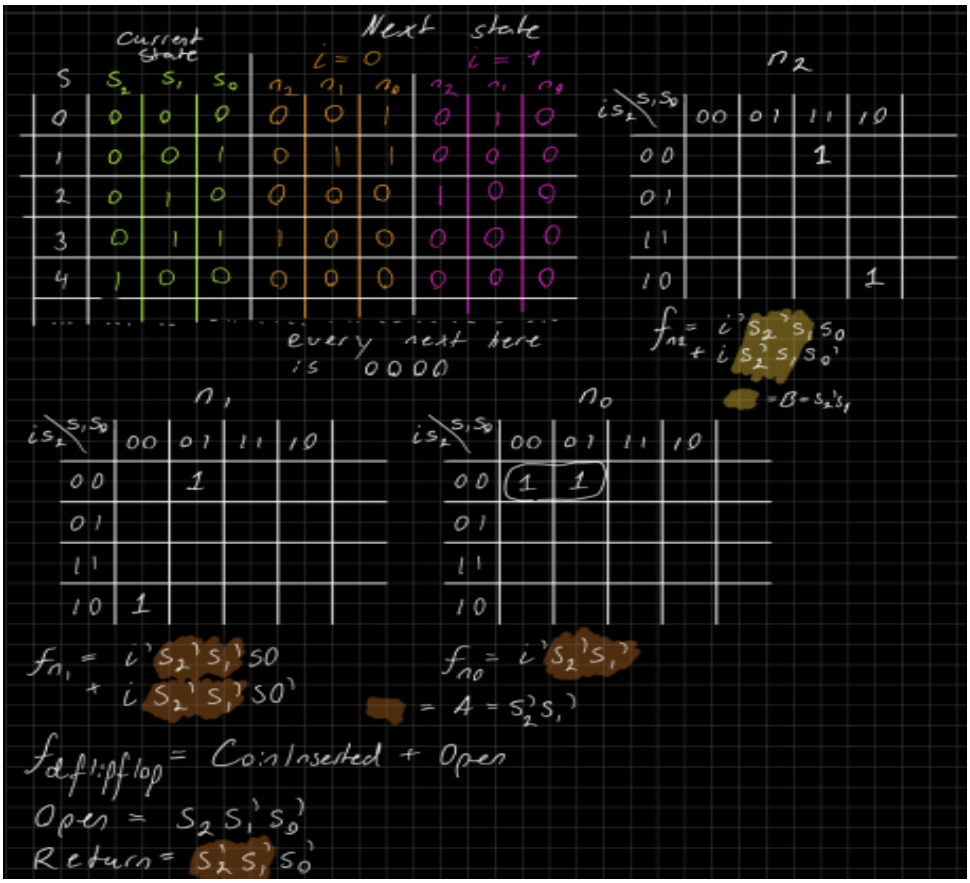
Consider that the load-enable of the D flip flops is active (set to "1") when a coin is inserted (input signal coin-inserted = 1) or the FSM output Open is "1" (at the final state), otherwise it is "0".

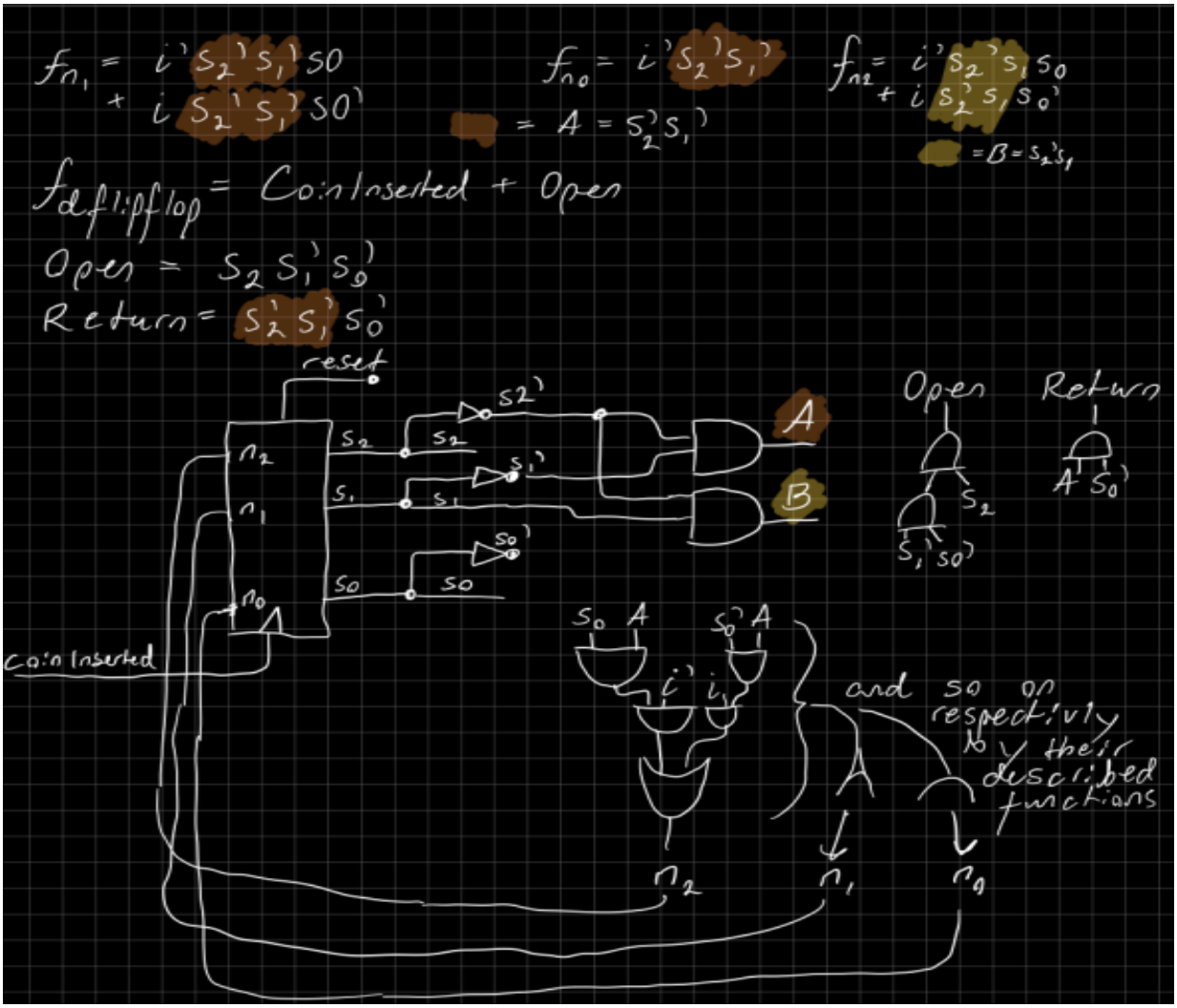
Note that two coins cannot be received at the same time. In addition, when the correct amount is collected, no more coins can be received until the machine resets again.

Make the state diagram and state table, choose a state assignment and implement the FSM deriving the Boolean functions for next state bits and output bits. Drawing the gatelevel circuit using D-flip-flops.

Answer
Similar to the following considering coins of 2 and 3 SEK and total of 6 SEK.


i = 0
i = 1
0kr
S0
return-1
2kr
S1
3kr
S2
4kr
S3
6kr
S4
open-1
i = x


current state
Next state
i = 0
i = 1
every next here is 0000
$f_{n2} = i' S_2' S_1 S_0 + i S_2' S_1 S_0' = B \cdot S_2' S_1$
$f_{n1} = i' S_2' S_1' S0 + i S_2' S_1' S0'$
$f_{n0} = i' S_2' S_1' = A = S_2' S_1'$
$f_{dff:ffIop} = CoinInserted + Open$
$Open = S_2 S_1' S_0'$
$Return = S_2' S_1' S_0'$



Question 7 Asynchronous Sequential circuits: (10 points)

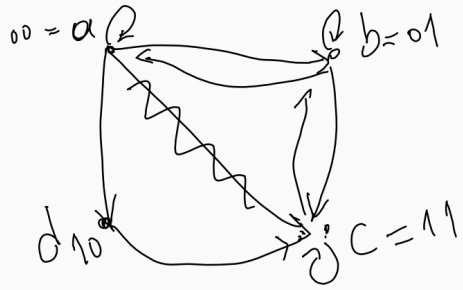
- Explain what is a critical race condition in an Asynchronous Sequential circuit and why it should be avoided.
- Select a state assignment for the Asynchronous Sequential circuit described by the following state table and make the necessary adjustments to ensure it is race-free, where x1, x2 are inputs of the circuit.

	Next State			
Current state:	x1x2= 00	x1x2= 01	x1x2= 11	x1x2= 10
a	b	a	c	b
b	b	a	a	c
c	b	c	c	c

Answer

Similar to Lecture on Asynchronous Circuits, slide 25.

	00	01	10	11
a	b	a	d	b
b	b	a	a	c
c	b	c	c	c
d	–	–	c	–



Question 8 Testing: (10 points)

- What is the use of a scan chain in a sequential circuit?
- Redraw the FSM gatelevel implementation of Question 6 including a scan chain. If you have not solved Question 6 draw add the scan chain to a generic Mealy FSM with 3 D Flip-Flops and undefined combinational logic parts which are shown as black boxes.

Answer

See lecture on designs and Test, slide 72. Scan chain should be added to the flip flops.

Question 9 Memory: (10 points)

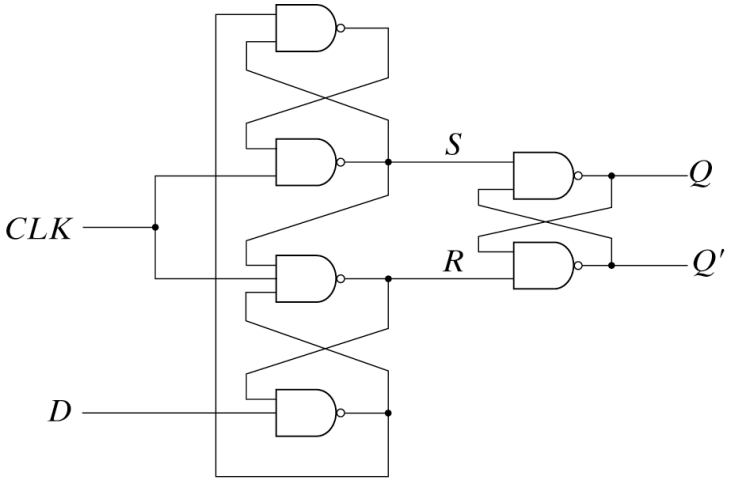
Describe the basic structure of a DRAM chip and the three operations used to access (read) a block of data in DRAM.

Answer

Lecture on Memories and Interconnects, slide 40 (best explained in the lecture recording).

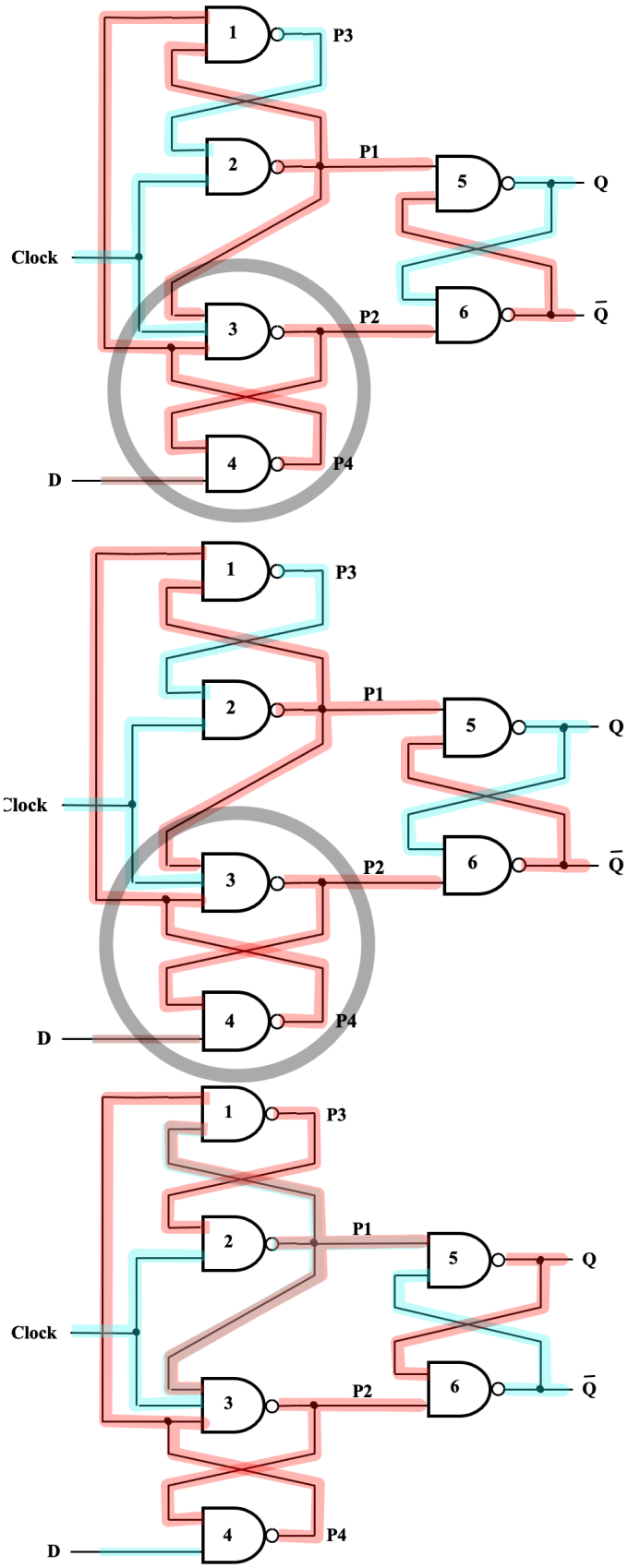
Question 10 Sequential Circuits and Timing: (10 points)

Explain how a positive edge triggered Flip-flop (shown in the figure bellow) works by explain the values of the NAND gates when clock is 0 and when clock switches to 1.



Answer

- Lecture on sequential circuits slides 15-17 (explained better in the recording of the lecture)


1
P3
2
P1
5
Q
Clock
3
P2
6
Q̄
4
P4
D
1
P3
2
P1
5
Q
Clock
3
P2
6
Q̄
4
P4
D
1
P3
2
P1
5
Q
Clock
3
P2
6
Q̄
4
P4
D

END of EXAM