

Final Exam

Explanation

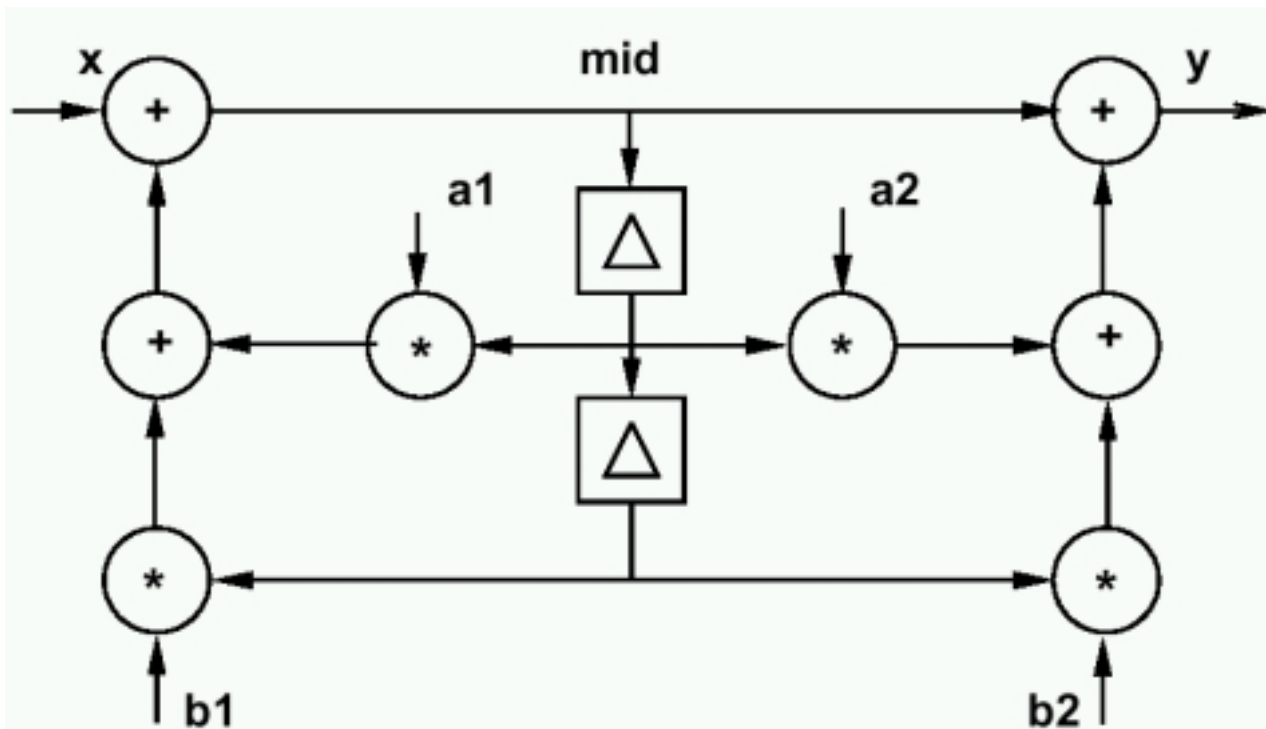
1. Pick 5 problems and solve them. Solving correctly 5 problems is sufficient to obtain the grade of A.
2. If you have enough time, solve as many other problems as you want, but please define which are your main 5 problems.
3. I give partial credits for good ideas. I give more problems so that everybody who knows even only a part of material can be fairly evaluated.
4. I subtract only some credit for numerical errors. Ideas and deep understanding of main principles are more important to me.
5. Verify your solutions. If you have time, verify using several methods.
6. Underline or mark otherwise your final solution to each point.
7. Try to write explanations to your solutions and procedures that will be as detailed as possible. This will help me to grade you fairly. Drawing graphs without explanations or formulas without their reasons is of little value for me while grading. Show always the sequence of design steps and explain each of your design choices. For instance, what was your motivation to select one type of a controller and not another type.
8. It is reasonable to start from easy problems and proceeds towards more time consuming problems.

Problem 1. Fast Transforms and Butterflies.

- (a) Draw a kernel of a Fast Reed-Muller Transform. Explain on which formula of Boolean Algebra it is based.
- (b) Illustrate transformation of a symbolic 3 variable Karnaugh Map of Boolean Function to its Positive Polarity Reed-Muller Form using a butterfly based on the RM Kernel from point (a).
- (c) Repeat point (b) for function: $f_1 = AB + C$. Show logic values in all nodes of the graph. Next repeat point (b) for function $f_2 = A'B'C' \oplus A$ where symbol A' means negation of input variable A .
- (d) Draw a purely combinational realization of the Butterfly diagram from point (b).
- (e) draw a pipelined realization of the circuit from point (d), explain in your own words how it works, draw for this pipeline a timing diagram that is typical to illustrate operations of pipelines.

Problem 2. Pipeline design, retiming and a controller.

- (a) Start from circuit shown here. Design a pipelined circuit for this graph. Retime if necessary (we started this project in the class and we spent much time!).
- (b) Design the controller FSM for this pipeline. It should be optimized and verified.
- (c) Draw the Timing table for your pipeline
- (d) Use this table for verification of retiming.

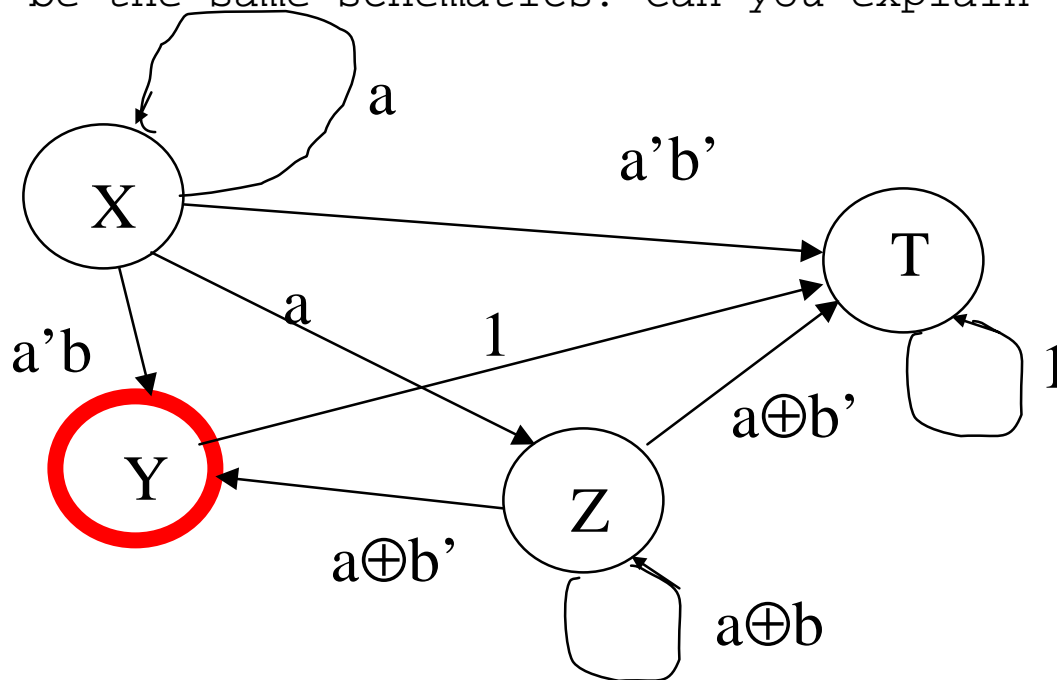


Problem 3. Design of parallel controllers.

- (a) Draw an arbitrary parallel flowchart that has some realistic (not necessarily practical) meaning, using parallel FORK and JOIN nodes. You can use other parallel nodes if you wish.
- (b) Illustrate realization of control using arbitrary method shown in the class.
- (c) Draw schematically a complete system of control unit and data path and analyze its behavior graphically (a timing diagram for data and most important controls) for one set of input data.

Problem 4. Reachability analysis.

- (a) Convert a non-deterministic FSM from Figure to an equivalent deterministic FSM using reachability analysis. Each state has its specific output with the same name, for instance being in state X is signaled by output X =1. Y is the accepting state.
- (b) Draw the schematics of the non-deterministic machine using D flip-flops
- (c) Draw the schematics of the deterministic machine using D flip-flops. Do not try to minimize the number of flip-flops
- (d) Can it be the same schematics? Can you explain why “yes” or why “not”?



Problem 5. Regular expressions.

- (a) Write a regular expressions of language **L** for the following event: *An even number of symbols **c** following symbol **b** or a divisible by three number of symbols **c** followed by an odd number of symbols **b**.*
- (b) Draw a graph of this regular expression
- (c) Convert the graph to a non-deterministic machine
- (d) Convert the non-deterministic machine to an equivalent deterministic machine.
- (e) Verification: For every possible sequence of letters a and b of length not larger than 3 **analyze** if it belongs to language L:
 - in the regular expression,
 - in the non-deterministic machine,
 - and in the deterministic machine.
- If it does not, what does it mean?

Problem 6. Iterative circuits.

- (a) Define what is an iterative circuit.
- (b) Write what is a relation between one-directional, one-dimensional iterative circuit and a Finite State Machine. Explain the Trade-off between speed and area in digital design and illustrate them on two versions of a circuit for comparison of two numbers - one iterative combinational and one a finite state machine.
- (c) Design an iterative combinational circuit with three outputs: $p = (A > B)$, $r = (A = B)$, $s = (A < B)$. Assume delay t_1 for every logic gate with 2 inputs. Compare starting from the least significant bit. Draw the transition graph (state machine) for the single combinational block. Calculate the total delay of the circuit. Draw the schematics. Explain your design stages.
- (d) Use the transition graph from point (c) above to draw the sequential FSM realizing serial comparison for the same task.

Problem 7. Turing Machine.

- Design of a Turing machine to calculate number $2n$ given number n on a tape. Both numbers are represented by subsequent ones.
- Example for $n=3$:
- $011100 \rightarrow 011101111100$
Initial head position Final head position

Perform the following:

- (a) draw the data path from functional blocks
- (b) draw the control unit and how the data path and the control unit are connected.
- (c) realize the control unit as any machine of your choice - Mealy, Moore, netlist of flip-flops, OR and branching gates, or a microprogrammed unit.
- (d) verify using your schematics and the example above ($n \rightarrow 2n$) that your machine works correctly.

Problem 8. Machines with stack.

- (a) Design a machine with a control unit, an input shift register and a stack that accepts language $\cup A^n B^n = AB \cup AABB \cup AAABBB \dots$
- (b) Show details of data path design.
- (c) Discuss the role of all signals.
- (d) Design a microprogrammed control unit for the stack and input register control.

Problem 9. Controller Design.

- (a) Design a sequential circuit with arbitrary blocks that executes operations of addition, multiplication, division and subtraction on *complex numbers*.
- (b) Assume that you have available blocks that realize combinational operations of addition, subtraction, multiplication and division of 8-bit registers.
- (c) Realize and draw the state graph of the controlling state machine and realize it using arbitrary method.
- (d) Draw the data path circuit. Show details of controlling registers. If necessary, optimize the controlling signals.
- (e) Verify your solution.

Problem 10. Minimization of Incompletely Specified Finite State Machines.

a	d/-	f/0
b	c/1	h/0
c	d/0	h/0
d	c/0	a/0
e	-/-	f/0
f	e/0	a/0
g	c/1	-/-
h	b/1	a/1

Machine M

- Given is Machine M
- (a) Find the minimal machine (in the number of states) that is equivalent to machine M
- (b) Draw the triangular table of machine M
- (c) Solve the triangular table
- (d) Find the maximal compatible groups of states
- (e) Solve graphically the covering/closure problem.
- (f) Formulate algebraically the binate covering problem.
- (g) Realize the machine using JK flip-flops and combinational gates.

Problem 11. Realization of Synchronous Finite State Machines.

a	d/-	f/0
b	c/1	h/0
c	d/0	h/0
d	c/0	a/0
e	-/-	f/0
f	e/0	a/0
g	c/1	-/-
h	b/1	a/1

- (a) Given is machine from the left. Realize this machine using D flip-flops and the excitation and output functions that would depend on the minimum total number of variables.
- (b) If you cannot minimize all these functions, try to minimize at least some and prove that you minimize them by some systematic method.
 - You do not have to prove that your solution is optimum but you must proceed rationally using the methods shown in class.
 - While solving this problem think about all FSM optimizing methods discussed in our class.
- (c) Using the final schematics demonstrate that you indeed minimized the number of arguments of some functions. Write specifically which ones. Prove with your comments that you understand the principles of state assignment and not only the procedure.

Problem 12. State Assignment of Synchronous Finite State Machines.

a b

	00	01	11	10
A	A	A	B	C
B	B	A	C	C
C	B	C	C	B
D	A	—	B	B

Transition table

a b

	00	01	11	10
A	01	11	11	11
B	01	01	1-	01
C	01	11	-1	01
D	01	--	-1	11

Output table

Given is machine M2 described with the following transition and output tables

- (a) Find Partitions determined by inputs, states and outputs. How they influence the final encoding. Check on the final circuit.
- (b) Find a Rule-based state assignment.
- (c) Find a Hypercube-based state assignment. You can use any way of creating the graph for embedding in hypercube.
- (d) Use the Multi-line method. Can you apply the basic theorem?
- (e) Draw the partition pairs and partition lattices and how they are useful in state assignment.
- (f) Can you combine and compare various methods for this example?
- Find a reasonable (not necessarily the best one) state assignment for JK flip-flops. Explain why this assignment is good for JK flip-flops. Illustrate with Kmaps of J and K excitation functions.

Problem 13. Scheduling and Register Allocation.

(a) How many registers to store the six variables (u1 to u6) ?

Inputs: v1, v2, v3, v4

Outputs: u5, u6

op1: u1 \leftarrow v1 + v2

op2: u2 \leftarrow v3 - v2

op3: u3 \leftarrow v3 + v4

op4: u4 \leftarrow u1 - u2

op5: u5 \leftarrow u2 + u3

op6: u6 \leftarrow u4 - u3

(b). Perform ASAP scheduling

(c). Perform ALAP scheduling

(d). Do life-time analysis of variables

(e). Assume that you have one adder and one subtracter, what is the minimum number of registers?

(f) Analyze three various variants of scheduling and allocation, not necessarily minimizing the number of registers or functional blocks, and compare their timings and costs.

(g). Select one variant and design both control unit and data path for it.

(h). Draw all necessary graphs. Are they compatibility or incompatibility graphs? Are they comparability graphs, chordal graphs or other graphs with some special properties? Explain the role of these graphs.

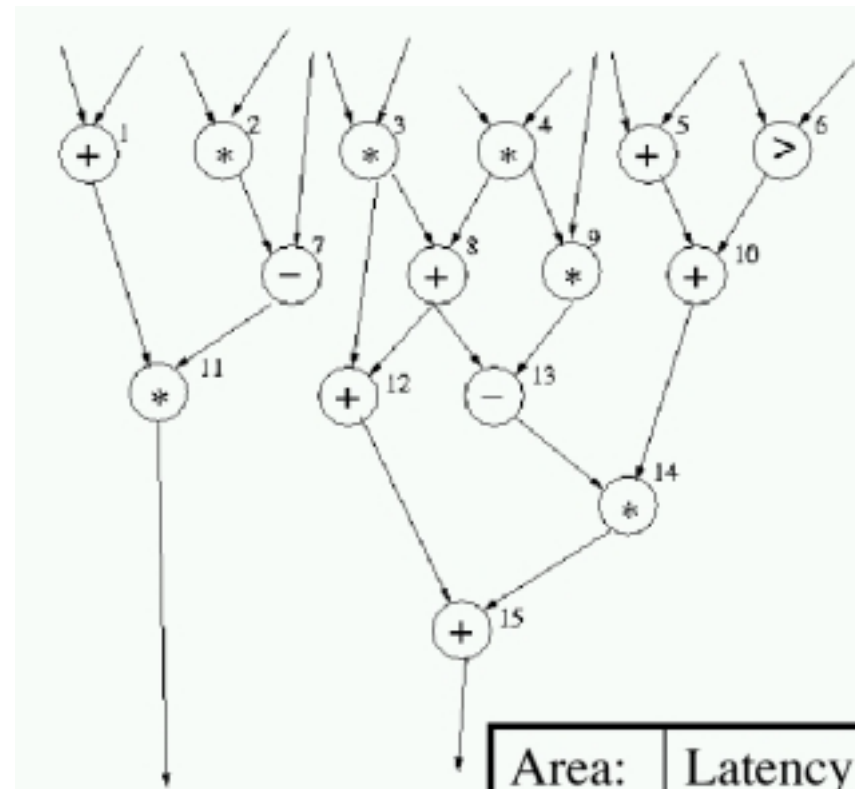
Problem 14

Schedule and allocate resources for each

operation, minimizing area and latency as much as possible.

- (a) Find the solution with the smallest area
- (b) Find the solution with the smallest latency.
 - In points (a) and (b) you are not required to give an optimal solution, since that may prove to be more difficult than can be done in a reasonable amount of time.
 - But you have to demonstrate that your reasoning is correct, you understand the problem and know at least some scheduling and allocation methods.

Unit	Size	Function
Multiplier	8	*
ALU	4	>, +, -
Comparator	2	>
Adder	2	+
Subtractor	2	-



Problem 15.

From Verilog to sequential logic circuit

```
module DIFFEQ (x, y, u , dx, a, clock, start);  
input [7:0] a, dx;  
inout [7:0] x, y, u;  
input clock, start;  
reg [7:0] xl, ul, yl;  
always  
  begin  
    wait ( start);  
    while ( x < a )  
      begin  
        xl = x + dx;  
        ul = u - (3 * x * u * dx) - (3 * y * dx);  
        yl = y + (u * dx);  
        @(posedge clock);  
        x = xl; u = ul ; y = yl;  
      end  
    endmodule
```

(a) Design the complete Data Path and Controller for this example. Any method from the class is applicable for any part of the complete design procedure.

(b) Explain your selections of methods and design decisions.

(c) Verify your solution.