UNIVERSITY OF CALIFORNIA, LOS ANGELES UCLA

BERKELEY · DAVIS · IRVINE · LOS ANGELES · RIVERSIDE · SAN DIEGO · SAN FRANCISCO SANTA BARBARA · SANTA CRUZ

CS M151B / EE M116C

Midterm Exam #1

All work and answers should be written directly on these pages, use the backs of pages if needed.

This is an open book, open notes quiz – but you cannot share books or notes.

We will follow the departmental guidelines on reporting incidents of academic dishonesty – do not make us enforce the rules. Keep your eyes on your own exam!

NAME: _____

ID: _____

Do not write anything in the area below on this page:

Problem 1: _____ (20)

Problem 2: _____ (40)

Problem 3: _____ (40)

Total: _____ (out of 100)

∞

1. ***Hardware Design ISA-bout Tradeoffs (20 points):*** Consider the following three properties:

   - *Instruction Count*
   - *Cycle Time (assume that the CPI of any particular instruction will stay the same)*
   - *Hardware Complexity* (i.e .the number of logic gates required to implement the hardware or the cost to verify the design - the hardware is more difficult to design/verify)

   We assume that the source code of the application we are going to evaluate does not change – but that it may be compiled differently based on the changes made below. For the following design decisions, indicate how these three properties could be impacted on the ***single cycle datapath*** from class. Your answer for each should be that it either ***could increase, could decrease, or will stay the same.*** You may only circle one answer per property. The first one is done for you as an example.

Example: ***Design Decision***: Use a carry lookahead adder instead of a ripple carry adder.			
• *Instruction Count*	could increase	could decrease	(will stay the same)
• *Cycle Time*	could increase	(could decrease)	will stay the same
• *Hardware Complexity*	(could increase)	could decrease	will stay the same

   a. ***Design Decision***: Eliminate a complex addressing mode from our ISA – all other addressing modes will remain. Instructions using the removed addressing mode will either be removed from the ISA or will use a simpler addressing mode.

• *Instruction Count*	(could increase)	could decrease	will stay the same
• *Cycle Time*	could increase	(could decrease)	will stay the same
• *Hardware Complexity*	could increase	(could decrease)	will stay the same

   b. ***Design Decision***: Extend our ISA to include multiply-add instructions which can do a multiply followed by an add in a single instruction.

• *Instruction Count*	could increase	(could decrease)	will stay the same
• *Cycle Time*	(could increase)	could decrease	will stay the same
• *Hardware Complexity*	(could increase)	could decrease	will stay the same

   c. ***Design Decision***: Hardware based on a register-memory machine instead of a load-store machine.

• *Instruction Count*	could increase	(could decrease)	will stay the same
• *Cycle Time*	(could increase)	could decrease	will stay the same
• *Hardware Complexity*	(could increase)	could decrease	will stay the same

2. ***Performance Anxiety (40 points)***: Suppose on a particular load-store machine-based processor there are four types of operations - loads, stores, branches, and ALU operations (i.e. adds, subtracts, multiplies). This will be implemented by some other datapath – **not** the single-cycle datapath we covered in class. Loads take 6 cycles, stores take 4 cycles, branches take 3 cycles, and ALU operations take 4 cycles. The architecture is not pipelined – so no instruction latency is overlapped – instructions are executed one at a time, in order. Answer the questions below – show your work.

   a. What is the CPI for an application with one million instructions that is 25% loads, 5% stores, 50% ALU operations, and 20% branches?

   CPI: 4.3

   $CPI = 6(0.25) + 4(0.05) + 3(0.2) + 4(0.5)$

   $= 4.3$

   b. The hardware running the application from part (a) has a cycle time of 312.5 ps (that is picoseconds). What is the execution time for that hardware to run the application?

   Execution Time $= 10^6 \times 4.3 \times 312.5 \times 10^{-12}$ ET: 1.344 msec

   c. Now suppose that we can eliminate 50% of all ALU operations through an algorithmic change (i.e. we save intermediate values in lookup tables). Unfortunately, this requires more loads – in fact, for every two ALU operations we eliminate, we need to add one load instruction. What is the new CPI?

   CPI: 4.63

   500,000 ALU ints

   ⇓

   250,000 ALU inst +125,000 Load Insts

   d. What is the execution time for the same hardware from part (b) to run the new application from part (c)?

   Exc Time $= (4.63) \times$

   $(875{,}000) \times$

   (312.5×10^{-12})

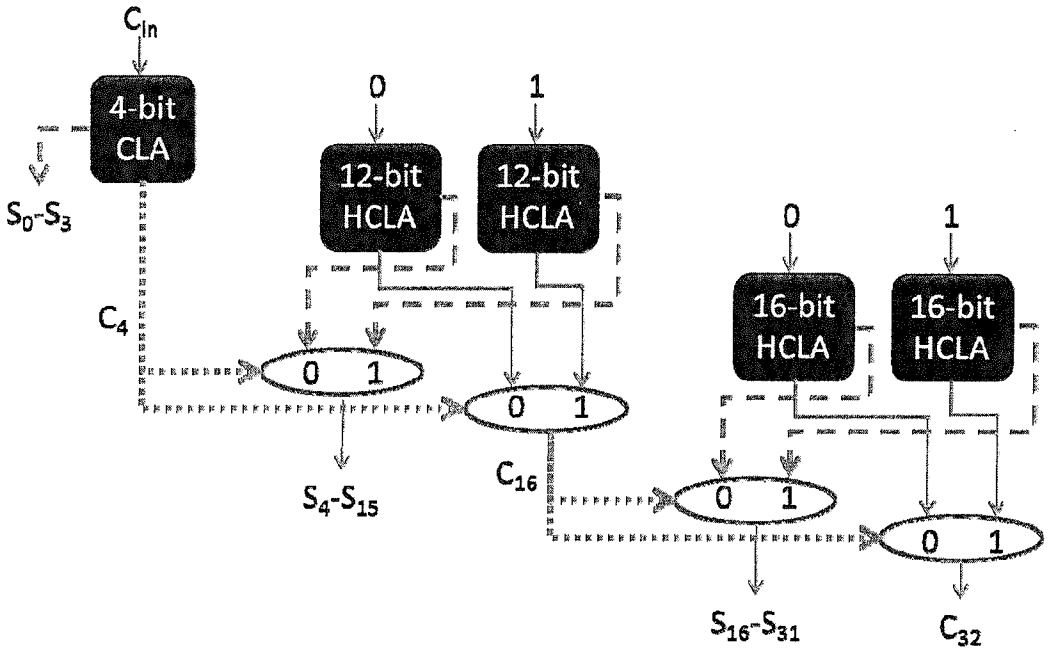
   ET: 1.266 msec

3. ***Putting the CLA into UCLA (40 points)***: Assume for the rest of this problem that all logic gates have the following delays:

Fan In	Delay
1	T
2	3T
3	4T
4	7T
5	9T
6 or more	2T x fan-in

So a 2-input AND gate would have delay 3T and a 4-input OR gate would have delay 7T. For simplicity, assume that mux's have delay 4T regardless of fan-in.

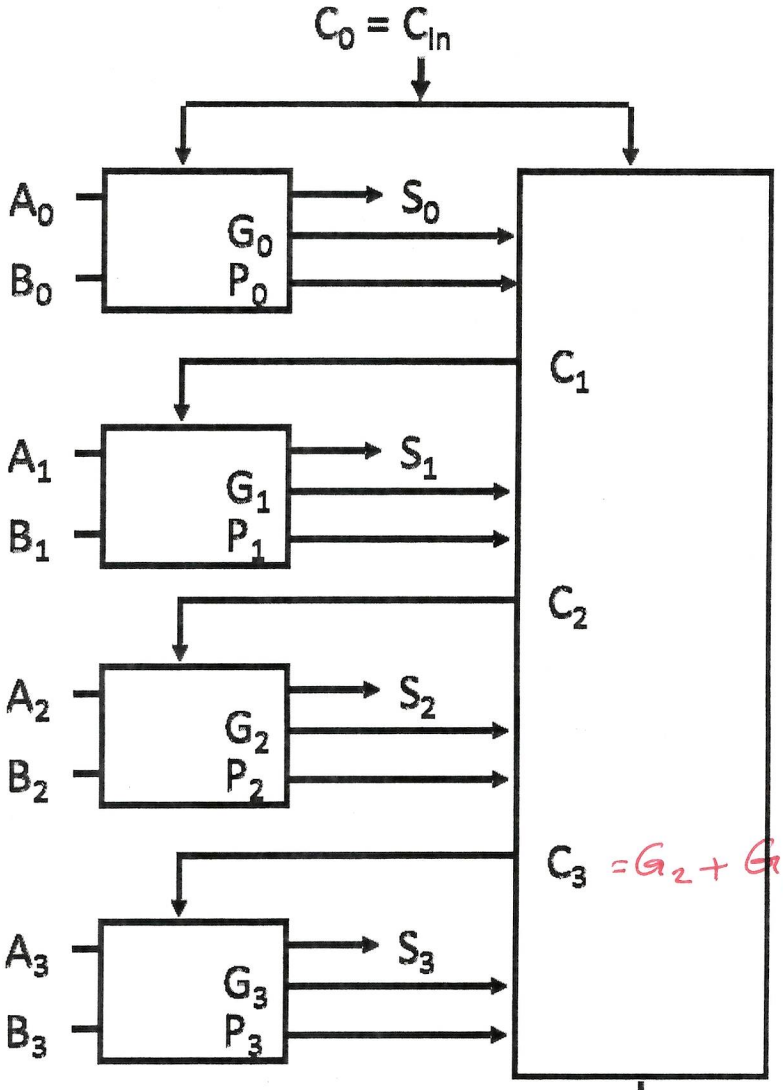
We will create a 32-bit adder out of some building blocks we've covered in class. We will use the 4-bit carry lookahead (4-bit CLA) that we covered in class as one basic building block of this design. And we will use it (as we did in class) to make a 16-bit hierarchical CLA (16-bit HCLA). And we will also use it to make a 12-bit hierarchical CLA (12-bit HCLA). But instead of connecting these in series to make a 32-bit adder, we will use carry select to speed up the 32-bit adder. The design will look as follows (be sure to note where we are using the 4-bit CLA and where we are using the 12-bit and 16-bit HCLAs):



Note that the 4-bit CLA is computing the sum of inputs A_0-A_3 and B_0-B_3 (i.e. producing sums S_0-S_3), the 12-bit HCLA is computing the sum of inputs A_4-A_{15} and B_4-B_{15} (i.e. producing sums S_4-S_{15}), the 16-bit HCLA is computing the sum of inputs A_{16}-A_{31} and B_{16}-B_{31} (i.e. producing sums S_{16}-S_{31}). The carry out of the 4-bit CLA selects the sums and carry out of the 12-bit HCLA. The carry out of the 12-bit HCLA selects the sums and carry out of the 16-bit HCLA. The 0 or 1 at the top of each HCLA is the hardwired C_{in}. Multiplexers are shown as ovals.

Your task is to find the maximal delay of this design – i.e. determine the delays of S_{0-31} and C_{32} – the maximal delay of these outputs will be the maximal delay of the entire design. To do this (and to help with possible partial credit) please use the diagrams on the following pages and fill in the tables in every page.

4-bit CLA:



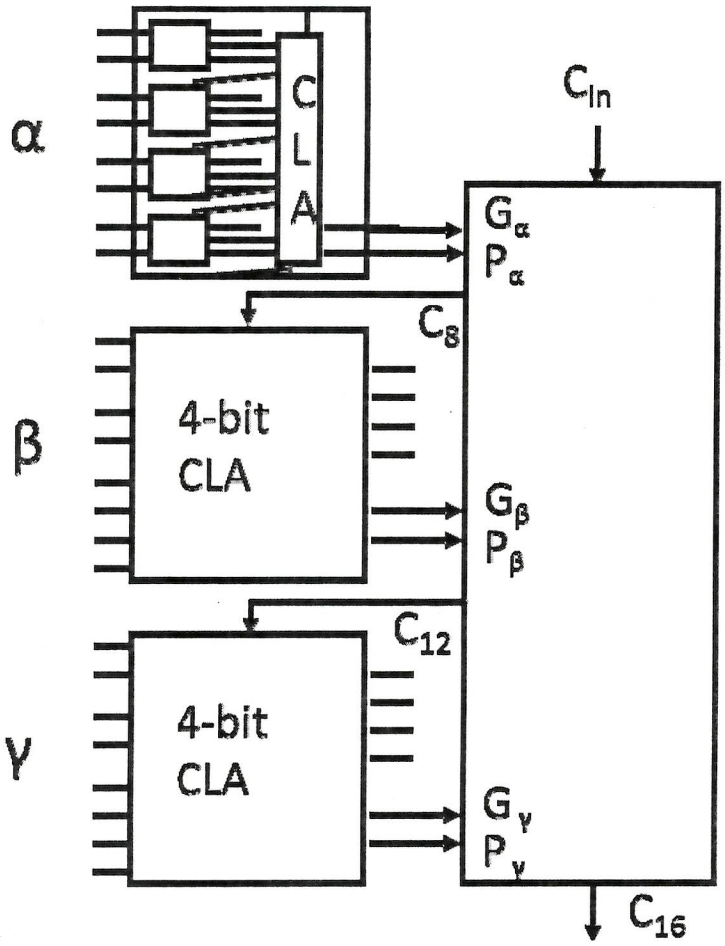
$C_3 = G_2 + G_1P_2 + G_0P_1P_2 + C_0P_0P_1P_2$

$C_4 = G_3 + G_2P_3 + G_1P_2P_3 + G_0P_1P_2P_3 + C_0P_0P_1P_2P_3$

4-bit CLA

Output	Delay (in terms of T)	
G0	3	*(2 points)*
P0	3	*(2 points)*
G3	3	*(2 points)*
P3	3	*(2 points)*
C3	17	*(2 points)*
C4 (just in this figure)	21	*(2 points)*
S3	21	*(2 points)*

12-bit HCLA:



12-bit HCLA

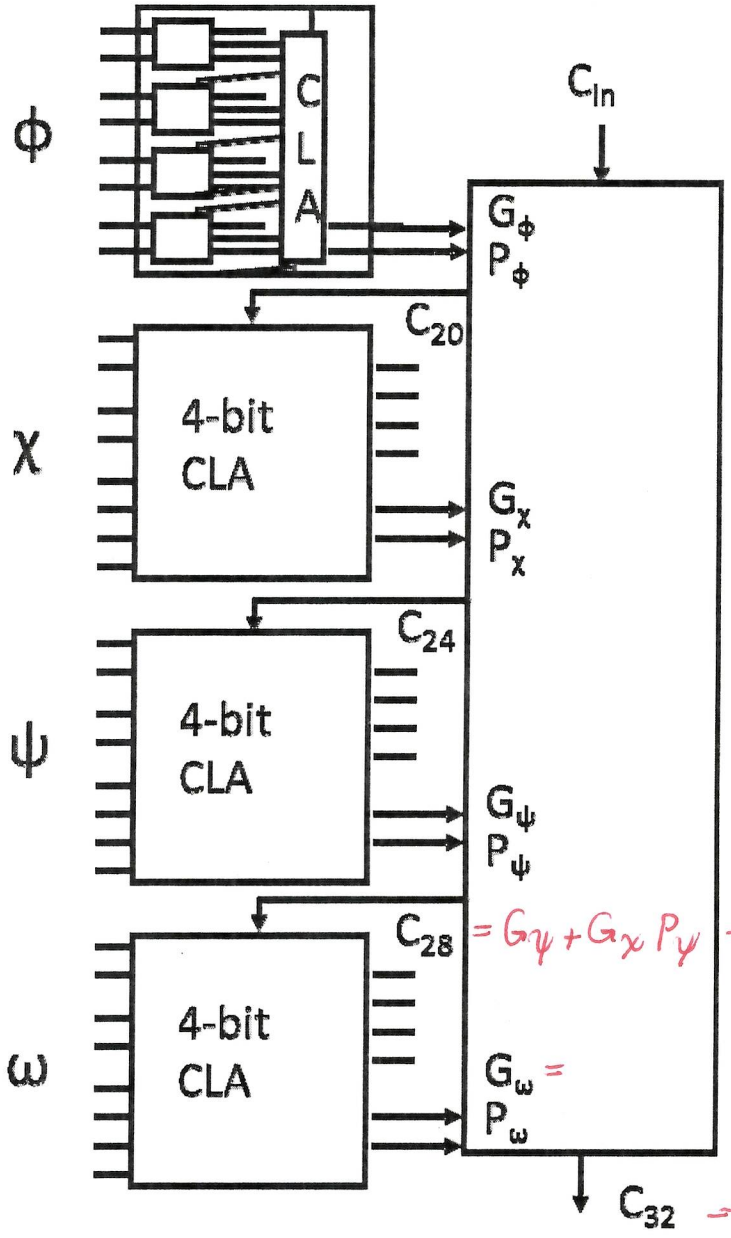
Output	Delay (in terms of T)	
Gα	17	*(2 points)*
Pα	10	*(2 points)*
C12	24	*(2 points)*
C16 (just in this figure)	28	*(2 points)*
S15	42	*(2 points)*

$P_\alpha = P_0 P_1 P_2 P_3$

$G_\alpha = G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3$

$S_{15} = a_{15} \oplus b_{15} \oplus C_{15}$

16-bit HCLA:



$= G_\psi + G_\chi P_\psi + G_\phi P_\chi P_\psi + C_{in} P_\phi P_\chi P_\psi$

$P_\omega = P_{28} P_{29} P_{30} P_{31}$

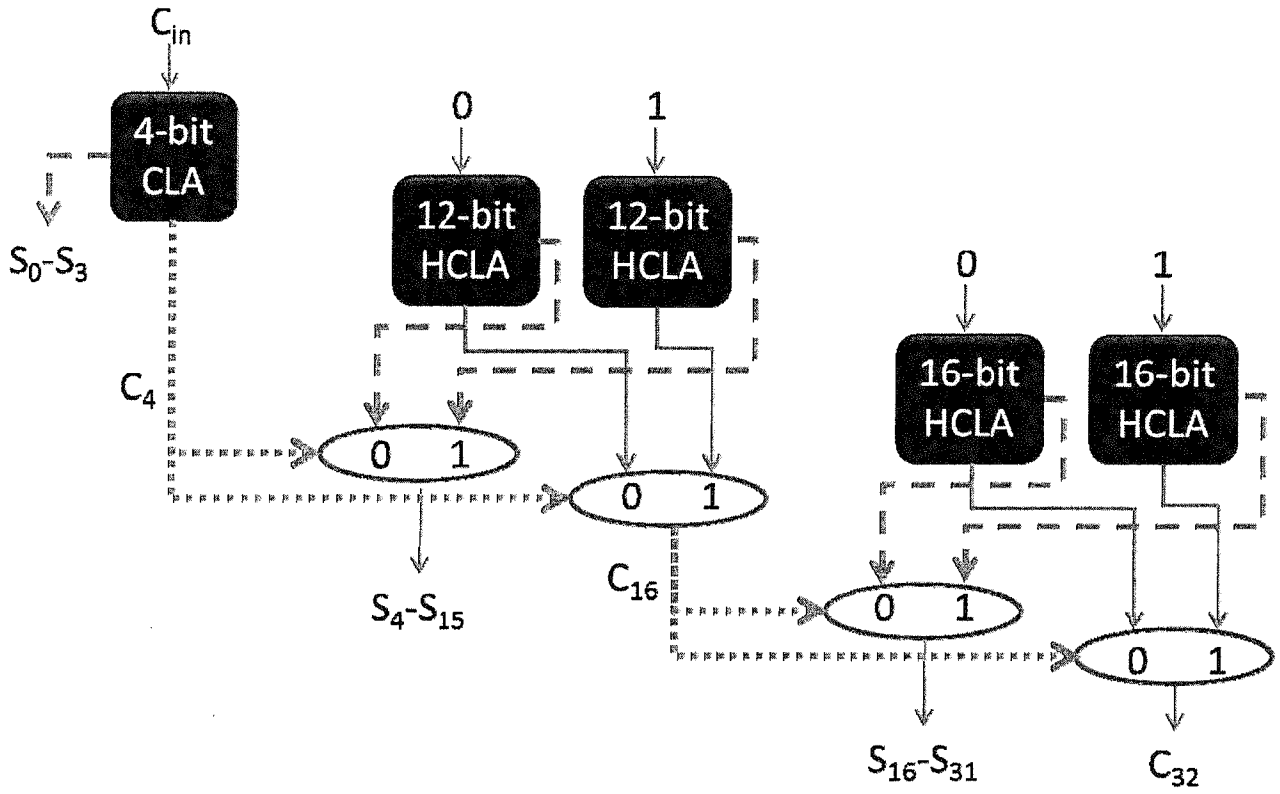
16-bit HCLA

$= G_\omega + P_\omega G_\psi + G_\chi P_\psi P_\omega + G_\phi P_\chi P_\psi P_\omega + C_{in} P_\phi P_\chi P_\psi P_\omega$

Output	Delay (in terms of T)	
Gω	17	*(2 points)*
Pω	10	*(2 points)*
C28	28	*(2 points)*
C32 (just in this figure)	33	*(2 points)*
S31	46	*(2 points)*

$S_{31} = A_{31} \oplus B_{31} \oplus C_{31}$

Entire Design:



Output	Delay (in terms of T)	
C16	32	*(2 points)*
C32	37	*(2 points)*

Find the maximum delay **in terms of T** of the 32-bit adder – take the maximum of all output bits – including the sum bits (S_0-S_{31}) and the final carry out (C_{32}).

Maximal Delay: 50 *(2 points)*