E4.43
AO1
ISE4.43

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2010

MSc and EEE/ISE PART IV: MEng and ACGI

SYNTHESIS OF DIGITAL ARCHITECTURES

Tuesday, 11 May 10:00 am

Time allowed: 3:00 hours

Corrected Copy
see p 5

There are THREE questions on this paper.

Answer ALL questions.

All questions carry equal marks

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible First Marker(s) : G.A. Constantinides
Second Marker(s) : C. Bouganis

Notation

The following notation is used in this examination paper.

- $\mathbb{N}$: the set of natural numbers.
- $\mathscr{P}(S)$: the power set of set S, *i.e.* the set of all subsets of S.

The Questions

1. a) Explain one advantage and one disadvantage of working with 'slicing floorplans' rather than 'general floorplans'. [3]

   b) Draw a two-dimensional floorplan layout consistent with the slicing tree shown in Fig. 1.1. [3]

   c) By constructing a different slicing tree also consistent with your two-dimensional floorplan layout, show that slicing trees are not canonical representations of the geometric structure. [4]

   d) Write the Polish expression corresponding to the skewed slicing tree representing your floorplan. [3]

   e) Show that by applying two operand swaps and one operator-chain complement, it is possible to produce the floorplan shown in Fig. 1.2. [4]

   f) When integer linear programming is used for floorplan optimization, explain two reasons why Manhattan distance is preferable to Euclidean distance. [3]

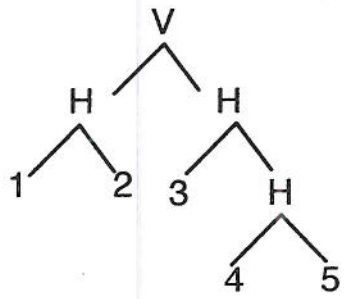


Figure 1.1 A slicing tree.

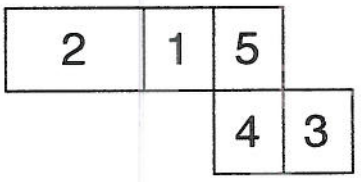


Figure 1.2 A floorplan.

2. This question compares two methods for calculating the reciprocal function $f(x) = 1/x$ in IEEE single precision floating point hardware. *Note: IEEE single precision floating point values are 32 bits long, and consist of 8 exponent bits, one sign bit, and 23 significand bits.*

In this question, you should take the area of one bit of ROM storage to be one unit. An n-bit integer adder/subtractor consumes $12n$ area units. An IEEE single precision floating point adder consumes 1536 area units, as does an IEEE single precision floating point multiplier.

a) A simple method is to use a ROM lookup table with a 32-bit input x and a 32-bit output $y \approx f(x)$. Calculate the area of the ROM. [2]

b) The ROM approach can be improved by breaking down x into its consituent parts $x = (-1)^s \times 1.m \times 2^{e-b}$, where s is the sign bit, $1.m$ is the mantissa, e is the exponent, and $b = 127$ is the bias. Design an alternative circuit where only the significand need be passed through a lookup table, resulting in a ROM with a 23-bit input and a 23-bit output, and calculate the resulting area of the overall circuit. To simplify your design, you may assume that m is never exactly zero. *Hint: Note that the number '1.m' denotes a real number with a '1' bit to the left of the radix (binary/decimal) point, and with the word stored in the significand to the right of the point. 1.m lies in the range* $[1, 2-2^{-23}] \approx [1,2]$. *Note also that the value e is stored as an 8-bit unsigned integer. This question uses the term 'mantissa' to denote the quantity* $1.m$, *while the term 'significand' is reserved for the quantity* m. [4]

c) A version of the design from part (b) with a controlled area-error tradeoff is also possible by only passing the most significant k bits of the significand through the lookup table. Thus the ROM has a k-bit input and a 23-bit output. Express the overall circuit area as a function of k. [2]

d) The operation of truncating all but k bits of significand creates a new floating-point number $\hat{x}$ from x, with value $\hat{x} = x(1+\delta)$, where $|\delta| \leq 2^{-k}$. The reciprocal actually calculated is then $\hat{y} \approx f(\hat{x}) \approx f(x)$. The result of the reciprocal can be written as $\hat{y} = y(1+\varepsilon)$, where $y = f(x)$. Find an upper bound (maximum value) for ε, the relative error in the computation, as a function of k. You may neglect the error induced inside the lookup table by rounding the final answer to an IEEE compliant representation. [3]

e) An alternative method of evaluation is to replace the lookup table in part (b) with a polynomial approximation $g(m) = \frac{2}{1.m}$ of a (scaled) reciprocal of the mantissa. Estimate the area of a the overall evaluation scheme based on a p-order Horner's scheme polynomial lookup, assuming no resource sharing and that all arithmetic is performed in IEEE single precision floating-point. [3]

Question continues overleaf...

f) For the scheme in part (e), we could use $g(m) = \frac{4}{3} - \frac{4}{9}(1.m)$. Let the relative error, $E(m)$, in the polynomial approximation be $E(m) = 1 - (1.m)g(1.m)$, *i.e.* we shall not consider finite precision effects. Calculate a bound on the worst-case value for $E(m)$. [3]

g) Is the first-order polynomial approximation scheme inferior, superior, or incomparable to the truncated table-lookup scheme? Justify your answer. [3]

3. a) Write a general ILP for solving the combined scheduling, resource binding, and module selection problem for a CDFG $G(V,E)$, to be executed using resource type set R, with type set function $T : V \to \mathscr{P}(R)$, resource cost c_r for a resource of type $r \in R$, delay d_r for a resource of type $r \in R$, where $d_{\min v} = \min_{r \in T(v)} d(r)$, $ASAP_v$ and $ALAP_v$ denote the ASAP and ALAP times of $v \in V$, respectively. You should use the variables $x_{vtir} = 1$ iff $S(v) = t$ and $Y(v) = (r,i)$, $b_{ir} = 1$ iff $\exists v \in V(Y(v) = (i,r))$.

Note that, in line with the notation used in the lecture notes, $S : V \to \mathbb{N}$ is the scheduling function, $Y : V \to \mathbb{N} \times R$ is the binding function, λ should be used to denote the deadline, and a_r should be used to denote an upper bound on the number of resources of type r.

[8]

b) Consider the code fragment { `a = b+c; d = a<e;` }, with $R = \{+, ALU\}$, $T(+) = R$, $c_+ = 1$, $c_{ALU} = 1.5$, $d_+ = 1$, $d_{ALU} = 2$, $\lambda = 4$. Completely enumerate possible choices of x_{vtir} variables. For each infeasible choice, state the constraint(s) violated. For each feasible choice, give the minimum value of the objective function when these variables are fixed. Hence identify an optimal solution. You may take $a_+ = 1$ and $a_{ALU} = 1$.

[8]

c) Given your results from part (b), carefully sketch the Pareto optimal frontier of area versus latency, indicating the essential features of the design at each Pareto optimal point.

[4]

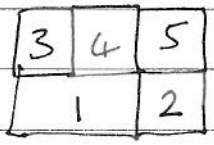
An ALU can perform an addition or comparison at any given time.

Synthesis of Digital Architectures
Solutions 2010

[E 4.43]
[Ise 4.43]
A01

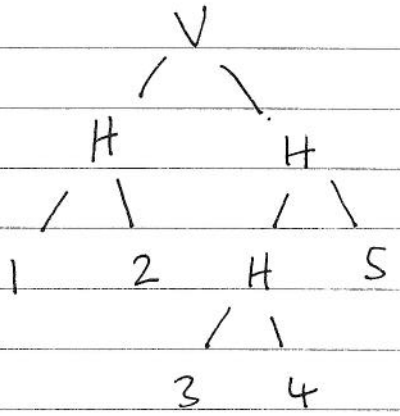
1. a) + Simple canonical representation, easy to search over heuristically.
- Limited: Slicing floorplans ⊂ floorplans. [3]

b)



[3]

c)



: same floorplan, hence lack of canonicity.

[4]

d) Above is SST. Polish expression is

12H 34H 5HV [3]

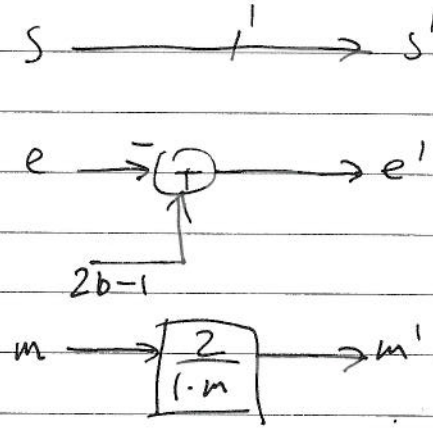
e) Swapping 1&2; 3&4, complementing HV gives

21H 43H 5VH

This corresponds to Fig 1.2. [4]

f) (i) Corresponds to standard design rules allowing only vertical & horizontal cuts.
(ii) Manhattan distance is linear – Euclidean is nonlinear.
[3]

2. a) 32×2^{32} units [2]

b) 

$$\frac{1}{(-1)^s \times 1.m \times 2^{e-b}}$$

$$= (-1)^s \times \frac{2}{1.m} \times 2^{b-e-1}$$

(for $m \neq 0$).

Area = $2^{23} \times 23 + 12 \times 8$ units. [4]

c) Area = $2^K \times 23 + 12 \times 8$ units. [2]

d) $1+\epsilon = \frac{1}{1+\delta} \leq \frac{1}{1-2^{-K}}$

$$\Rightarrow \epsilon \leq \frac{2^{-K}}{1-2^{-K}}$$ [3]

e) Using Horner's scheme requires p adders and p multipliers with no resource sharing.

Area = 3072p. [3]

f) Worst case attained at $1.m = 1$ and $1.m = 2$.
$|\epsilon(x)| = 1/q$. [3]

g) for this area (3072p), we would allow up to $K = 7$. The table based scheme has worst case $\leq \frac{2^{-7}}{1-2^{-7}}$.

which is much smaller than $1/q$. Hence the first-order scheme is inferior. [3]

3. a) $\min : \sum_{r \in R} c_r \sum_{i=1}^{a_r} b_{ir}$

$$\forall v \in V, \quad \sum_{r \in T(v)} \sum_{i=1}^{a_r} \sum_{t=ASAP_v}^{ALAP_v - d_r - d_{min_v}} x_{vtir} = 1 \qquad (*)$$

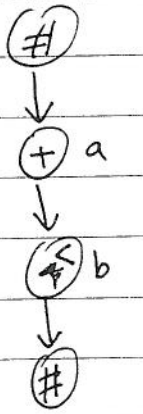
$$\forall t \in \Omega, \ \forall r \in R, \ \forall i \in \{1, 2, \ldots, a_r\} \qquad (+)$$

$$\sum_{v \in V : r \in T(v)} \sum_{t' \in \{t - d_r + 1, \ldots, t\} \cap \{ASAP_v, \ldots, ALAP_v - d_r + d_{min_v}\}} x_{vt'ir} \leq b_{ir}$$

$$\forall (v', v) \in \epsilon \quad \sum_{r \in T(v)} \sum_{i=1}^{a_r} \sum_{t=ASAP_v}^{ALAP_v - d_r + d_{min_v}} t \cdot x_{vtir} \geq \sum_{r \in T(v')} \sum_{i=1}^{a_r} \sum_{t=ASAP_{v'}}^{ALAP_{v'} - d_r + d_{min_{v'}}} (t + d_r) x_{vtir} \qquad (\text{if } \alpha)$$

(book work) [8]

b) CDFG



$ASAP_a = 0$ $\quad ALAP_a = 1$

$ASAP_b = 1$ $\quad ALAP_b = 2$

I will choose $a_+ = 2$, $a_{ALU} = 1$ but other choices are valid.

Variables: $x_{a,0,1,+}^{\rightarrow x}$; $x_{a,1,1,+}^{\rightarrow y}$; $x_{a,0,1,ALU}^{\rightarrow z}$ if [illegible]

$x_{b,1,1,ALU}^{\rightarrow p}$; $x_{b,2,1,ALU}^{\rightarrow q}$; $b_{1,+}$; $b_{1,ALU}$

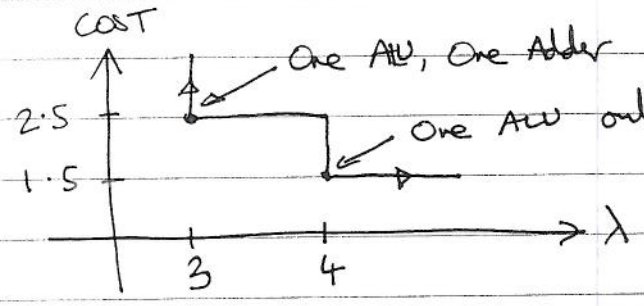
Complete enumeration overleaf. [8]

Complete enumeration: I have left out lines impossible w.r.t. (*)

x	y	z	p	q	
1	0	0	1	0	Cost = 2·5
0	1	0	1	0	→ Impossible w.r.t. (**)
0	0	1	1	0	→ Impossible w.r.t. (†)
1	0	0	0	1	Cost = 2·5
0	1	0	0	1	Cost = 2·5
0	0	1	0	1	Cost = 1·5 ← OPTIMAL.

[8]

c)



[4]