IMPERIAL COLLEGE LONDON

E4.43
AO1
ISE4.43

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2007

MSc and EEE/ISE PART IV: MEng and ACGI

Corrected Copy

SYNTHESIS OF DIGITAL ARCHITECTURES

Thursday, 3 May 10:00 am

Time allowed: 3:00 hours

There are FIVE questions on this paper.

Answer THREE 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

SYNTHESIS OF DIGITAL ARCHITECTURES

Notation

The following notation is used in this exam paper.

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

The Questions

1. In this question, the set of statements defined in some nested loop code is denoted V. The set of iterations of the loop nest, the *iteration space*, is denoted $\mathscr{I}$. Throughout the question, execution time of loop index calculations may be neglected, and all statements take a single cycle to execute.

   a) Discuss the relative merits of the following alternative scheduling procedures for nested-loop code:

      i) Unrolling the loops and using a list-scheduling algorithm on the resulting straight-line code,

      ii) Imposing an affine-by-statement form on the scheduling function $S : V \times \mathscr{I} \rightarrow \mathbb{N}$.

   [4]

   b) Consider the code shown in Figure 1.1.

      i) Express $\mathscr{I}$ in the form $\mathscr{I} = \{i | Ai \leq b \text{ and } i \in \mathbb{Z}^n\}$, for some matrix A and vector b.

      [4]

      ii) Identify any data dependences present in this code, and write the corresponding scheduling constraints in terms of iteration vector i and affine-by-statement parameter t, where the scheduled start time of iteration i is given by $t^T i$.

      [4]

      iii) Use the *vertex method* to identify appropriate additional constraints (linear in t), bounding the execution time below a further variable. Hence write a linear program to schedule this code.

      [5]

      iv) Suggest a solution to this linear program and hence state a scheduling function explicitly.

      [3]

```
for $i_1$ = 2 to 10
    for $i_2 = i_1$ to 12 - $i_1$
        s[$i_1$][$i_2$] = s[$i_1$-1][$i_2$] + s[$i_1$][$i_2$-1]
    end
end
```

Figure 1.1 Some nested loop code

2. In this question, you are asked to derive a methodology for resource binding a set of identical operations V. You are given a pre-defined conflict graph $G(V,E)$ which is not necessarily an interval graph. No register sharing is performed in this question.

a) Explain the relationship between graph colouring and resource binding a resource dominated circuit.

[5]

b) Using an appropriate set of binary decision variables, $\{x_{vc}\}$, with the interpretation $x_{vc} = 1$ iff operation $v \in V$ has colour $c \in \mathbb{N}$, formulate an ILP for colouring $G(V,E)$ using the minimum number of colours.

[6]

c) Suggest appropriate functions to model the growth of multiplexer area and combinational delay with the number of data inputs, assuming all data inputs are of an identical word-length.

[3]

d) Hence modify your ILP so that it can be used to minimize the silicon area (resources + multiplexers) of a resource binding, given both a conflict graph $G(V,E)$ and an upper-bound S on the worst-case combinational delay introduced by resource-sharing multiplexers.

[6]

3. A significant problem in architectural synthesis is to quickly estimate the capacitance of wires given a netlist, without any prior information on physical placement on silicon. Capacitance is approximately proportional to wirelength in VLSI design. Each wire has one *source* and one or more *destinations* on the chip. The *bounding box* of a wire is the smallest rectangle on the chip that encloses both the source and all destinations of a wire, and that has vertical and horizontal sides (*i.e.* its sides are parallel to the axes). The *fanout* of a wire is the number of destinations. Typically the source and destinations of a wire are approximated as the respective centres of their (rectangular) source and destination resources.

a) Half the perimeter of the bounding box is often used as a measure of wirelength. Informally justify this measure by example, and suggest any possible shortcomings. (*Hint: Consider separately examples of fanout 1, 2, and 3*).

[4]

b) Let R denote the set of resources and $E \subseteq \mathscr{P}(R)$ denote a set of nets (wires) connecting resources such that $\{r_1, r_2, \dots, r_n\} \in E$ iff resources r_1, r_2, …, r_n are connected via a single net. Let a subset $R' \subseteq R$ of resources have fixed x and y centre coordinates determined *a priori*, given by $x : R' \rightarrow \mathbb{R}$ and $y : R' \rightarrow \mathbb{R}$, respectively. Let the width and height of the resources be given by $w : R \rightarrow \mathbb{R}$ and $h : R \rightarrow \mathbb{R}$, respectively. Formulate a floorplanning mixed integer linear program involving the variables listed in Table 3.1,with the objective to minimize the total length of wiring in the design.

[6]

c) Comment on how well the mixed-ILP methodology matches the stated requirement 'to quickly estimate the capacitance of wires given a netlist'.

[3]

d) To simplify the estimation problem, it has been proposed to remove the 'no overlap' constraints in the floorplan. Write the correspondingly simplified mathematical program, and comment on the worst-case solution time for this modified program.

[4]

e) Comment on the accuracy of the resulting estimate.

[3]

Table 3.1. Floorplanning variables.

x_i	The x-coordinate of the centre of resource $i \in R$.
y_i	The y-coordinate of the centre of resource $i \in R$.
δ_{ij}, η_{ij}	Binary variables encoding relative placement of resources $i \in R$ and $j \in R$
g_e	The coordinate of the rightmost edge of the bounding box enclosing $e \in E$.
q_e	The coordinate of the leftmost edge of the bounding box enclosing $e \in E$.
a_e	The coordinate of the topmost edge of the bounding box enclosing $e \in E$.
s_e	The coordinate of the bottommost edge of the bounding box enclosing $e \in E$.

4. You are designing a special-purpose co-processor. The co-processor reads a data value $x \in [0, \pi/2]$ from an input, calculates the expressions $y = x * \cos(x) - x$ and $z = (x + \sin(x)) - 3$, and writes the outputs y and z. Multiplication takes two cycles, whereas addition, subtraction, external reads and writes ~~all take a single~~ cycle. The entire behaviour must complete within 5 cycles. Additional reference material for this question is given in Table 4.1.

ARE zero-

a) Draw a CDFG for this behaviour, leaving the function calls as unexpanded **F** nodes.

[3]

b) Perform an ASAP and ALAP schedule for all nodes in this CDFG, expressing your answers in terms of parameters d_1 and d_2, corresponding to the unknown delays of the cos and sin functions, respectively.

[4]

c) If the functions are to be approximated by polynomial evaluation using Horner's scheme, determine the greatest feasible order of the two polynomials.

[4]

d) Hence determine appropriate coefficients of maximal-accuracy (in the uniformly weighted least-squares sense) polynomials $f(x) = c_0 + c_1 x + \ldots + c_n x^n$ for the two functions, and the corresponding CDFGs in the lower level of CDFG hierarchy. Show all your working.

[7]

e) Is the greatest feasible polynomial order under Horner's scheme equal to the greatest feasible order regardless of evaluation scheme? Justify your answer.

[2]

Table 4.1. Some Orthogonal Polynomials over $[-1, 1]$.

Chebyshev-I	Legendre
$\phi_i(x) = 2^{i-1} \prod_{k=1}^{i} \left\{ x - \cos\left[\frac{(2k-1)\pi}{2i}\right] \right\}.$	$\phi_i(x) = \frac{1}{2^i i!} \frac{d^i}{dx^i} \left\{ (x^2 - 1)^i \right\}.$

5. On a modern field-programmable gate array (FPGA), there is a mixture of fine-grain logic elements, typically small ROMs known as lookup tables (LUTs), and dedicated circuitry for performing the common operations of multiplication and data storage (RAMs). These resource types can be expressed as $R = \{L, M, S\}$, for LUT, Multiplier, and Storage, respectively. This question concerns the mapping of CDFGs consisting of addition, multiplication, and memory nodes onto such a technology.

Associated with a CDFG $G(V,E)$ is a type function $T : V \rightarrow \mathscr{P}(R)$ that indicates which possible FPGA components could implement each given node. Addition can be implemented only by LUTs, while multiplication can be implemented by either LUTs or multipliers and data storage can be implemented by LUTs or RAMs.

The delay of a node $v \in V$ when implemented using resource type $r \in R$ is denoted d_{vr}. The area of a node $v \in V$ when implemented using resource type $r \in R$ is denoted a_{vr}.

Only ASAP scheduling, and no resource sharing should be performed in this question.

a) Using binary decision variables $\{x_{vr}\}$, with the interpretation $x_{vr} = 1$ iff node $v \in V$ is implemented using resource type $r \in R$, formulate an ILP to model the combined ASAP scheduling and module selection for this problem. The objective should be to minimize the overall latency of the CDFG, and a feasible solution must use no more than total area A (a given constant).

[9]

b) FPGAs are *configurable* devices, meaning that they should be able to implement any one of a *set* of CDFGs. The same resource can be used to implement different nodes in different CDFGs, but no resource sharing should be performed *within* a CDFG. A set of CDFGs $\{G_1(V_1,E_1), G_2(V_2,E_2), \ldots, G_k(V_k,E_k)\}$ is given. Formulate (and explain) an ILP to find the appropriate allocation of silicon area to each of the three component types in order to minimize the worst case overall latency across all CDFGs while using no more than a total area A (a given constant).

[8]

c) Given that $a_{vr} \leq a_{vr'}$ for all v whenever $r = M$ and $r' = L$ and also whenever $r = S$ and $r' = L$, is it ever possible that a minimum-area FPGA resulting from the preceding formulation has $x_{vr} = 1$ for some $v \in V$ and $r = L$? Justify your answer.

[3]

Synthesis of Digital Architectures E4.43

Solutions 2007 ISE4.43

AO1

Q1

a) (i) ⇒ large (exponential) increase in code size ⇒ large run time, poor heuristic quality (depending on heuristic)

(ii) ⇒ restricted functional form ⇒ can't expose all //ism.

(BOOKWORK) [4]

b) (i) $I = \{x \mid Ax \leq b \wedge x \in \mathbb{Z}^n\}$ (NEW COMPUTED EXAMPLE)

with $A = \begin{pmatrix} -1 & 0 \\ +1 & 0 \\ +1 & -1 \\ +1 & +1 \end{pmatrix}$ $b = \begin{pmatrix} -2 \\ 10 \\ 0 \\ 12 \end{pmatrix}$

[4]

~~c)~~ Two dependences

(ii) $\begin{pmatrix} i_1 \\ i_2 \end{pmatrix} \leftarrow \begin{pmatrix} i_1 - 1 \\ i_2 \end{pmatrix}$ $\begin{pmatrix} i_1 \\ i_2 \end{pmatrix} \leftarrow \begin{pmatrix} i_1 \\ i_2 - 1 \end{pmatrix}$

$\Downarrow$ $\Downarrow$

~~$\forall \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} \in \mathbb{Z}$~~

$(1 \ \ 0)t \geq 1$ (1) $(0 \ \ 1)t \geq 1$ (2)

[4]

Q1

~~d)~~ (iii) Vertices at $\begin{pmatrix}2\\2\end{pmatrix}$, $\begin{pmatrix}6\\6\end{pmatrix}$, $\begin{pmatrix}10\\2\end{pmatrix}$

So
$$(2 \quad 2)t + 1 \leq \lambda \qquad (3)$$
$$(6 \quad 6)t + 1 \leq \lambda \qquad (4)$$
$$(10 \quad 2)t + 1 \leq \lambda \qquad (5)$$

L.P. is then

min: λ

s.t. (1) – (5). [5]

~~e)~~ (iv) $t = \begin{pmatrix}1\\1\end{pmatrix}$

$$s(i) = (1 \quad 1)\, i$$
$$= i_1 + i_2 \, .$$
[3]

Q 2

a) Nodes ⟺ Operations (BOOKWORK)

Edges ⟺ Resource conflicts

Colours ⟺ Resources

Graph colouring ⟺ Min # resources [5]

b) Need at most $|V|^2$ variables – one colour per node max. (NEW APPLICATION OF THEORY)

$$\min \sum_{c=1}^{|V|} y_c$$

s.t. $\forall \{v_1, v_2\} \in E \quad \forall c \in \{1, \dots, |V|\}$,

$$x_{v_1c} + x_{v_2c} \leq 1 \qquad (1)$$

$\forall v \in V \quad \forall c \in \{1, \dots, |V|\}$

$$y_c \geq x_{vc} \qquad (2)$$

[6]

c) $\text{Area}(n) = K_1 n$ (NEW COMPUTED EXAMPLE)

$\text{Delay}(n) = K_2 \log_2 n$

[3]

Q2

(NEW APPLICATION OF THEORY)

d) We require $k_2 \log_2 n \leq S$

$\Rightarrow n \leq 2^{S/k_2}$, a constant.

So:

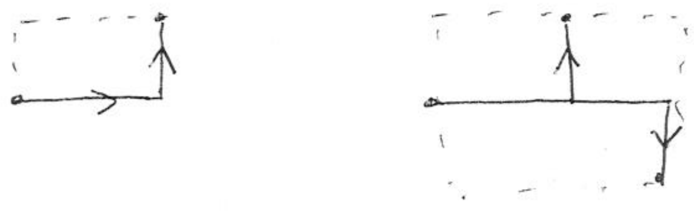
$$\min : \sum_{c=1}^{|V|} \left(y_c + \overset{\text{(two inputs).}}{2} k_1 \sum_{v \in V} x_{vc} \right)$$

s.t. (1) – (2)

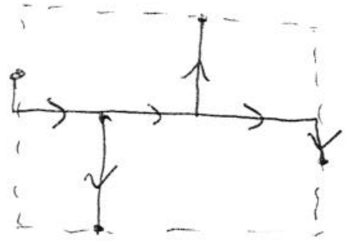
$$\forall c \in \{1, \ldots, |V|\} \quad \sum_{v \in V} x_{vc} \leq 2^{S/k_2} \qquad (3)$$

[6]

Q3. a) $\frac{1}{2}$ perimeter is min Manhattan wirelength for fanout = 1 or 2 : (NEW THEORY)



This is not the case for fanout 3 or more



[4]

b) Min: $\sum_{e\in E} (g_e - q_e + a_e - s_e)$ (NEW APPLICATION OF THEORY)

s.t. $\forall e \in E\ \forall i \in e$
$$\begin{cases} g_e \geq x_i \\ q_e \leq x_i \\ a_e \geq y_i \\ s_e \leq y_i \end{cases} \quad (1)$$

$$\forall i \in R\ \forall j \in R \begin{cases} M\delta_{ij} + M\eta_{ij} + x_i - x_j \geq \frac{1}{2}(w(i)+w(j)) \\ M\delta_{ij} + M(1-\cancel{8}\eta_{ij}) + x_j - x_i \geq \frac{1}{2}(w(i)+w(j)) \\ M(1-\delta_{ij}) + M\eta_{ij} + y_i - y_j \geq \frac{1}{2}(h(i)+h(j)) \\ M(1-\delta_{ij}) + M(1-\eta_{ij}) + y_j - y_i \geq \frac{1}{2}(h(i)+h(j)) \end{cases} \quad (2)$$

$\delta_{ij} \in \{0,1\}$, $\eta_{ij} \in \{0,1\}$

5

$$\forall i \in R' \quad \begin{cases} x_i = x(i) \\ y_i = y(i) \end{cases} \qquad (3)$$

[6]

c) Badly, as # integer vars is $\Omega(|R|^2)$ and MILP run time is $\Omega(2^{\#\text{vars}})$.

(NEW COMPUTED EXAMPLE) [3]

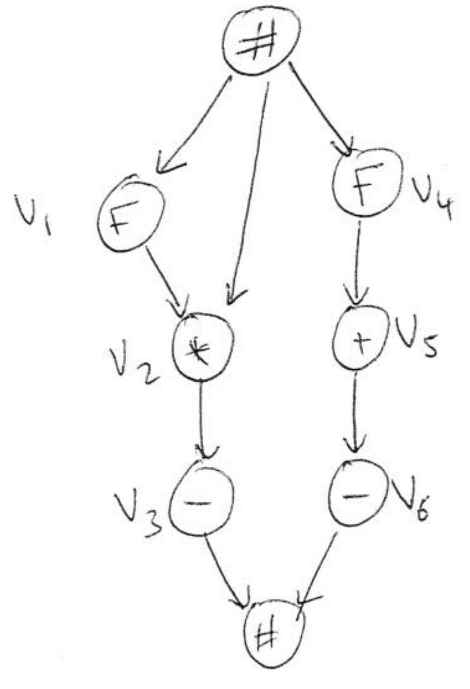
d) Remove (2) & all integer vars. Now a pure LP $\Rightarrow$ poly run time.

(NEW APPLICATION OF THEORY) [4]

e) If widths & heights are small compared to fixed x & y coordinates, likely to be accurate (spaced-out).

(NEW THEORY) [3]

Q4 a)



(NEW COMPUTED EXAMPLE)

[3]

b)

	ASAP	ALAP
V_1	0	$2-d_1$
V_2	d_1	2
V_3	d_1+2	4
V_4	0	$3-d_2$
V_5	d_2	3
V_6	d_2+1	4
SOURCE	0	$\min(5-d_1, 6-d_2)$
SINK	$\max(d_1+3, d_2+2)$	5

(NEW COMPUTED EXAMPLE) [4]

c) Horner's scheme:

order	min. cycles
0	0
1	3
2	6

(NEW APPLICATION OF THEORY)

Require $\max(d_1+3,\ d_2+2) \leq 5$

$$\Rightarrow d_1 \leq 2 \quad \& \quad d_2 \leq 3$$

Max order for $\sin = 1$
$\cos = 0$

[4]

d) Uniformly weighted $\Rightarrow$ Legendre (NEW COMPUTED EXAMPLE)

$x \in [0, \pi/2]$

$$\Rightarrow u = \frac{4x}{\pi} - 1 \quad \in [-1, 1]$$

$$\langle \phi_0, \phi_0 \rangle = \int_{-1}^{1} 1 \, du = 2$$

$$\langle \phi_1, \phi_1 \rangle = \int_{-1}^{1} u^2 \, du = \frac{1}{3}u^3 = \frac{2}{3}$$

(i) $\sin(x)$

$$\left\langle \sin \frac{\pi(u+1)}{4}, \phi_0 \right\rangle = \int_{-1}^{1} \sin \frac{\pi(u+1)}{4} \, du$$

$$= -\frac{4}{\pi} \cos\left(\frac{\pi(u+1)}{4}\right) \Big|_{-1}^{1}$$

$$= -\frac{4}{\pi}(0-1)$$

$$= \frac{4}{\pi}$$

$\langle \sin \frac{\pi(u+1)}{4}, \phi_1 \rangle$

$$= \int_{-1}^{1} u \sin \frac{\pi(u+1)}{4}\, du$$

$$= -\frac{4u}{\pi} \cos \frac{\pi(u+1)}{4} + \int_{-1}^{1} \frac{4}{\pi} \cos \frac{\pi(u+1)}{4}\, du$$

$$= -\frac{4}{\pi}(0+1) + \left(\frac{4}{\pi}\right)^2 \sin \frac{\pi(u+1)}{4} \Big|_{-1}^{1}$$

$$= -\frac{4}{\pi} + \left(\frac{4}{\pi}\right)^2$$

$$= \frac{4}{\pi}\left(\frac{4}{\pi} - 1\right).$$

$$a_0 = \frac{4}{\pi} / 2 = \frac{2}{\pi}$$

$$a_1 = \frac{4}{\pi}\left(\frac{4}{\pi} - 1\right) \Big/ \left(\frac{2}{3}\right)$$

$$= \frac{6}{\pi}\left(\frac{4}{\pi} - 1\right)$$

so $f(x) = \frac{2}{\pi} + \frac{6}{\pi}\left(\frac{4}{\pi} - 1\right) u$

$$= \frac{2}{\pi} + \frac{6}{\pi}\left(\frac{4}{\pi} - 1\right)\left(\frac{4x}{\pi} - 1\right)$$

$$= \underbrace{\frac{2}{\pi} - \frac{6}{\pi}\left(\frac{4}{\pi} - 1\right)}_{} + \underbrace{\frac{6}{\pi}\left(\frac{4}{\pi} - 1\right)\frac{4}{\pi}\, x}_{}$$

$$C_0 = \frac{2}{\pi}\left(1 - \frac{12}{\pi} + 3\right) \qquad C_1 = \underline{\underline{\frac{24}{\pi^2}\left(\frac{4}{\pi} - 1\right)}}$$

$$= \frac{2}{\pi}\left(4 - \frac{12}{\pi}\right) = \frac{8}{\pi}\left(1 - \frac{3}{\pi}\right)$$

(ii) <u>cos</u>

$$\left\langle \cos \frac{\pi(u+1)}{4}, \phi_0 \right\rangle$$

$$= \int_{-1}^{1} \cos \frac{\pi(u+1)}{4} \, du$$

$$= \frac{4}{\pi} \sin \frac{\pi(u+1)}{4} \Big|_{-1}^{1}$$

$$= \frac{4}{\pi}$$

$$\Rightarrow a_0 = \frac{4}{\pi} / 2 = \frac{2}{\pi}$$

$$f(x) = \underbrace{\frac{2}{\pi}}_{\underline{\underline{c_0}}} .$$

[7]

e) While in general they may not be equal, e.g. Estrin's method, in this example they are. No re-arrangement of a degree 0 or degree 1 polynomial can speed-up its' execution.

(NEW THEORY) [2]

Q5

a) Let us denote the sink node as v_*.

Then (NEW APPLICATION OF THEORY)

$$\min: \; S_{v_*}$$

$$\text{s.t.} \quad \forall (v_1, v_2) \in E \quad S_{v_2} \geq S_{v_1} + \sum_{r \in T(v_1)} d_{vr} x_{vr}$$

$$\forall v \in V \quad \sum_{r \in T(v)} x_{vr} = 1$$

$$\sum_{v \in V} \sum_{r \in T(v)} a_{vr} x_{vr} \leq A$$

[9]

b) We will use binary variables x_{vr} for $v \in V_1 \cup V_2 \cup \cdots \cup V_K$ and $r \in R$.

(NEW APPLICATION OF THEORY)

$$\min: \; \lambda$$

$$\text{s.t.} \quad \forall i \in \{1, \ldots, n\} \quad S_{v_{*K}} \leq \lambda$$

$$\forall i \in \{1, \ldots, n\} \; \forall (v_1, v_2) \in E_i \quad S_{v_2} \geq S_{v_1} + \sum_{r \in T(v_1)} x_{vr}$$

$$\forall i \in \{1, \ldots, n\} \; \forall v \in V_i \quad \sum_{r \in T(v)} x_{vr} = 1$$

$$\forall i \in \{1, \ldots, n\} \quad \sum_{v \in V_i} \sum_{r \in T(v)} a_{vr} x_{vr} \leq A$$

[8]

c) Yes, because although LUT-based nodes are larger, they could potentially be shared across more ~~bench~~ CDFGs. For exple, when one CDFG contains multiplicatn but another doesn't.

(NEW THEORY) [3]