IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2008

MSc Analogue Systems

Corrected Copy

SYNTHESIS OF DIGITAL ARCHITECTURES

Friday, 9 May 10:00 am

Time allowed: 3:00 hours

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

This page is intentionally left blank.

SYNTHESIS OF DIGITAL ARCHITECTURES

Notation

The following notation is used in this exam paper.

- $\mathbb{N}$ is the set of natural numbers.
- R is the resource type set.

The Questions

1.   This question concerns the application of suitable scheduling and binding algorithms to construct a datapath for the code shown in Fig. 1.1, which represents one iteration of a method for solving systems of linear equations. This behaviour is to be implemented using no more than four adder/subtractors, four multipliers, and one divider, all of which take one cycle to execute.

      a)   Construct a CDFG $G(V, E)$ for this code, give each node a unique label, and annotate the CDFG with the corresponding line number of the code. Note that in construction of the CDFG, no assumptions on commutativity or associativity of operations should be made, and no common sub-expressions should be eliminated.

[5]

      b)   Name a suitable heuristic to schedule the CDFG, and derive the corresponding schedule function $S : V \to \mathbb{N}$. Carefully show your working.

[5]

      c)   Derive a register conflict graph (or an equivalent interval diagram) for the scheduled CDFG, under the assumptions that all results are read at some undetermined future time, and that no registers are required for external inputs. State the graph's chromatic number and its clique number.

[5]

      d)   Derive an appropriate binding function $Y : V \to R \times \mathbb{N}$, and state the number of resources required by this binding.

[5]

```
function CGiter(a11,a12,a21,a22,x1,x2,r1,r2,p1,p2)
begin
    normsq := r1*r1 + r2*r2;                                              Line 0
    alpha := normsq / ((p1*p1)*a11 + (p1*p2)*(a21+a12) + (p2*p2)*a22);    Line 1
    x1new := x1 + alpha*p1;                                               Line 2
    x2new := x2 + alpha*p2;                                               Line 3
    r1new := r1 - alpha*(a11*p1 + a12*p2);                                Line 4
    r2new := r2 - alpha*(a21*p1 + a22*p2);                                Line 5
    beta := (r1new*r1new + r2new*r2new)/normsq;                          Line 6
    p1new := r1new + beta*p1;                                             Line 7
    p2new := r2new + beta*p2;                                             Line 8
    return(x1new,x2new,r1new,r2new,p1new,p2new);                         Line 9
end
```

Figure 1.1 An iteration of the Conjugate Gradient algorithm.

2. In new technologies, it is often the case that wires take a significant amount of time to transfer data, and this impacts on optimal scheduling decisions for a DFG. This question concerns building an ILP formulation for scheduling which considers this issue.

a) First, formulate an ILP for an area-constrained scheduling problem, without regard for physical placement. Use the symbols shown in Fig. 2.1. *Hint: Base your formulation on the optimal scheduling ILP studied in lectures, but use an area constraint rather than constraints on the number of each resource type. Assume the circuit is resource dominated.*

[10]

Symbol	Meaning
x_{vt}	Binary variable. 1 iff node v scheduled at time t.
a_r	Variable. Bound on the number of resources of type r used in the schedule.
R	Constant. Resource type set.
V	Constant. Set of nodes to schedule.
$ASAP_v$ ($ALAP_v$)	Constants. ASAP (ALAP) time of node v.
d_v	Constant. Latency (delay) of node v.
$T(v)$	Constant. Type of node v.
c_r	Constant. Area cost of one instance of resource type r.
A	Constant. Bound on total area of design.
v_z	Constant. The sink node.
λ	Constant. Bound on the sink-node schedule time.

Figure 2.1 Notation for Question 2.

We shall approximate the physical placement of resources by a simple model. The chip is divided into four quadrants. Registers are physically placed next to the resource writing the result stored in that register; no register sharing is in operation. Multiplexers at the inputs to resources are physically placed next to the appropriate resource. Communication on wires within a quadrant takes zero time, whereas communication between quadrants takes one cycle. Each quadrant has a maximum area of $A/4$.

b) By replacing the binary decision variables with $\{x_{vtq}\}$, with the interpretation $x_{vtq} = 1$ iff node v is scheduled at time t and physically placed in quadrant q, extend the ILP above to the case where each resource needs to be allocated to one of the four quadrants, this time taking wire delay into account. Clearly list the variables and constraints in your formulation, and explain their meaning. *Hint: The linear constraints on binary variables $\varepsilon \geq \eta_1 - \eta_2$ and $\varepsilon \geq \eta_2 - \eta_1$ together ensure that $\varepsilon \geq 1$ if $\eta_1 \neq \eta_2$.*

[10]

This page is intentionally left blank.

3. This question concerns retiming, and a possible extension for non-resource-dominated circuits. In this question, multipliers, adders, dividers, reads, and writes are combinational nodes with propagation delay of 1 unit, 1 unit, 3 units, 0 units, and 0 units, respectively.

a) Explain the initialization problem in retiming with reference to an example of a combinational circuit where the initialization problem may occur.

[2]

b) Construct a delay-weighted DFG to represent the inner-loop of the code shown in Fig. 3.1, using node types '+', '*', '/', 'r' (read), and 'w' (write).

[3]

The retiming formulation given in lectures takes no account of register area, yet retiming decisions impact on register size. Assuming all data words are of equal size, the clock period / register area tradeoff can be explored by minimizing a linear combination of these two objectives, *i.e.* min : $T_{clk} + \alpha N$, where N is the total number of words stored in registers in the design, and $\alpha > 0$ is a parameter.

c) Formulate an ILP that would allow the exploration of this tradeoff in the manner described, assuming all nodes with output produce a single word.

[10]

Under two different values of α, the different delay-weighted DFGs shown in Fig. 3.2 are obtained.

d) Derive pseudo-code corresponding to these two delay-weighted DFGs, and comment on the relative magnitude of α_1 and α_2.

[5]

```
while( true )
begin
    read x;
    y = x2 + z2;
    q = y * 3;
    z = 5 / q;
    write q;
    x2 = x1;
    x1 = x;
    z2 = z1;
    z1 = z;
end
```
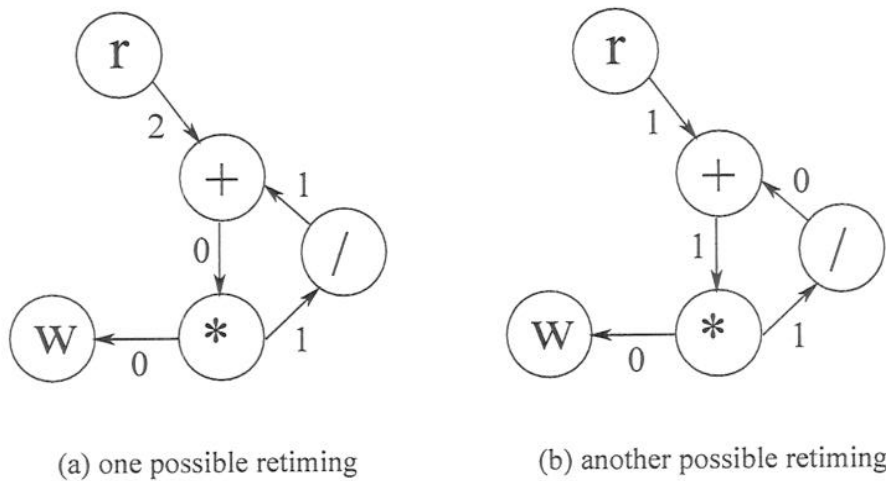
Figure 3.1 Original code, before retiming.

(a) one possible retiming   (b) another possible retiming

Figure 3.2 Delay-weighted DFGs under parameter (a) α_1 and (b) α_2.