E4.43
AO1
ISE4.43

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2005

MSc and EEE/ISE PART IV: MEng and ACGI

Corrected Copy

SYNTHESIS OF DIGITAL ARCHITECTURES

Monday, 16 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) : K. Masselos

© University of London 2005

This question relates to the code for the function **func**, shown below, where the function **exp**(x) returns the value e^x.

```
func( a, b, c )
{
  return (a + 63*b) + exp(c);
}
```

To implement this behaviour, you have the resource set R = {add, mult, ROM}, with area costs of 1, 2, and 10 units area, and delays of 1, 2, and 1 cycle, respectively. You may also use the definition of the Legendre polynomials over the range [-1,1] given in (Equation 1.1).

$$\phi_i(x) = \frac{1}{2^i i!}\frac{d^i}{dx^i}\left\{(x^2-1)^i\right\} \quad \text{(Equation 1.1)}$$

a) Draw the CDFG for function **func**, leaving any function calls un-expanded.

[3]

b) Assuming the **exp(.)** function is implemented as a ROM lookup, perform an ASAP scheduling on your CDFG. State the scheduled start time of each node, and the overall latency.

[3]

c) Perform a resource binding on your scheduled CDFG. State the resulting binding function, and calculate the resulting area of the circuit. State any assumptions you make.

[4]

d) Given that the parameter **c** is known to lie in the range [-1, 0], construct a 2nd order uniformly-weighted least-squares polynomial approximation to **exp(c)**, $g(x) = a_0 + a_1x + a_2x^2$. State the values of a_0, a_1, and a_2.

[5]

e) Apply strength reduction to the code, and construct the modified CDFG of the strength-reduced code, once the call to **exp(c)** has been replaced by a Horner's scheme evaluation of the appropriate approximation.

[2]

f) Perform an ASAP scheduling and resource binding on the modified CDFG. State the start time of each node, and the resulting minimum area of the circuit.

[3]

2

The notation in Table 2.1 applies to this question.

Table 2.1

Symbol	*Meaning*
V	vertex set of the CDFG
E	edge set of the CDFG
R	resource type set
S	scheduling function
Y	binding function
x_{vtir}	binary decision variable with $x_{vtir} = 1$ iff $S(v) = t$ and $Y(v) = (r,i)$
b_{ir}	binary decision variable with $b_{ir} = 1$ iff instance i of resource type r is used
a_r	an upper bound on the number of resources of type $r \in R$
ASAP_v	the ASAP start time of node $v \in V$ when all nodes have minimum latency
ALAP_v	the ALAP start time of node $v \in V$ when all nodes have minimum latency
$T(v)$	the resource type set for node $v \in V$
c_r	the cost of resource type $r \in R$
d_r	the delay of resource type $r \in R$
$d_{\text{min } v}$	the minimum delay of node $v \in V$

a) Formulate the minimum-cost latency-constrained combined scheduling, binding, and module selection problem as an ILP in the variables $\{x_{vtir}\}$ and $\{b_{ir}\}$.

[5]

b) Construct the CDFG for the code shown in Figure 2.2.

[1]

c) By substituting appropriate values into your formulation from part (a), show that the ILP for this CDFG, when subject to an overall latency constraint of 2 cycles, simplifies to (Figure 2.2). Ensure you clearly show how each inequality is formed. (You may make the assumptions shown in Table 2.4)

[8]

d) By enumeration of the possible variable values, or otherwise, derive all feasible solutions to the ILP shown in Figure 2.3, and identify the optimum.

[6]

[Question 2 continues on the next page]

```
p = x + 2;      // operation a
q = x + x;      // operation b
r = p * 2;      // operation c
```

Figure 2.2

minimize: $4b_{1,\text{mult}} + 2b_{1,\text{ripple}} + 3b_{1,\text{lookahead}}$
subject to:
$x_{a,0,1,\text{lookahead}} = 1$
$x_{b,0,1,\text{ripple}} + x_{b,0,1,\text{lookahead}} + x_{b,1,1,\text{lookahead}} = 1$
$x_{c,1,1,\text{mult}} = 1$
$x_{a,0,1,\text{lookahead}} + x_{b,0,1,\text{lookahead}} \leq b_{1,\text{lookahead}}$
$x_{b,1,1,\text{lookahead}} \leq b_{1,\text{lookahead}}$
$x_{b,0,1,\text{ripple}} \leq b_{1,\text{ripple}}$
$x_{c,1,1,\text{mult}} \leq b_{1,\text{mult}}$

Figure 2.3

Table 2.4

Parameter	*Value*
R	{mult, ripple, lookahead}
c_{mult}	4
c_{ripple}	2
$c_{\text{lookahead}}$	3
d_{mult}	1
d_{ripple}	2
$d_{\text{lookahead}}$	1
a_{mult}	1
a_{ripple}	1
$a_{\text{lookahead}}$	1
$T(v)$	{mult} if v is a multiplication {ripple, lookahead} if v is an addition

3

This question concerns the code shown below. The behaviour is to be implemented using the resource types: multiplier, subtractor. A multiplier takes two cycles to execute, ~~an adder~~ subtractor takes one cycle. Both read and write operations take no cycles.

You may take a comparator to be a zero-latency operation.

```
func(p,q) {
 if( p > 0 )
  return (q*p)*(1 - p);
 else
  return 0;
}

main( ) {
 read r;
 read x;
 for i = 1 to N {
  a = func(x,r);
  x = 2*a;
 }
 write x;
}
```

a) Draw the CDFG for this code, representing read and write calls as read and write nodes, respectively.

[3]

b) Perform an ASAP scheduling on this CDFG. For nodes with more than one execution time, express the times in terms of i and/or N, as appropriate.

[3]

c) Construct the resource conflict graphs for the scheduled CDFG, and colour them to obtain a resource binding.

[2]

d) Construct the register conflict graph for this scheduled CDFG, and state whether the graph is an interval graph (you may exclude the variable i from consideration).

[3]

e) Colour the register-conflict graph using 4 colours.

[3]

f) By reference to clique-number, show that the register-conflict graph cannot be coloured using 3 colours.

[3]

g) Draw the complete datapath resulting from your scheduling, resource binding, and register sharing (you may exclude reads and writes).

[3]

4

This question concerns the following code, to be implemented using adder/subtractor resources, which take one cycle each to execute.

```
p = x + y;  // operation a
q = p – 2;  // operation b
r = p + 2;  // operation c
s = q – y;  // operation d
t = q + y;  // operation e
u = r – x;  // operation f
v = x + x; // operation g
```

a) Construct the CDFG for this code, labelling all the vertices with their operation letter.

[2]

b) What is the minimum overall latency required to execute this code?

[1]

c) Name an appropriate algorithm to schedule this code under a given upper bound on the number of adder/subtractors to be used.

[1]

d) Schedule the code using the algorithm from part (c), for all positive integer values of the upper bound. Break ties in urgency by choosing the operation in order of the alphabet (e.g. **a** before **b**). State the start time of each operation for all values of the bound.

[3]

e) Name an appropriate algorithm to schedule this code given that it must complete within a given upper bound on the number of cycles.

[1]

f) Schedule the code using the algorithm from part (e), for all integer values of the bound greater than or equal to the minimum overall latency. Break ties in urgency by choosing the operation in order of the alphabet (e.g. **a** before **b**). State the start time of each operation for all values of the bound.

[3]

g) Draw an area/latency design-space for your behaviour, clearly identifying all the feasible schedules you have constructed from parts (d) and (f).

[2]

h) Define the term "inferior design", and identify any inferior designs in your design-space.

[1]

[Question 4 continues on the next page]

[Question 4 continued]

i) Is one of the two algorithms used more likely to produce inferior designs? Briefly explain your answer.

[3]

j) Define the term "Pareto-optimal". Identify any Pareto-optimal designs in your design-space.

[1]

k) Will one of the two algorithms always produce Pareto-optimal designs? Explain your answer.

[2]

5.

This question concerns the following scheduled and bound code, in which all operations take a single cycle to execute. **x** and **y** are the 8-bit inputs to this circuit, and **t4**, **t5**, and **t6** are the 16-bit outputs. The controller of this circuit consumes a significant proportion of the overall area.

```
t1 = x + y;    // cycle 0, resource (+,1)
t3 = x * y;    // cycle 0, resource (*,1)
t2 = t1 * t3;  // cycle 1, resource (*,1)
t4 = t1 + t3;  // cycle 1, resource (+,1)
t5 = t2 / t4;  // cycle 2, resource (/,1)
t6 = w * z;    // cycle 2, resource (*,1)
```

A register sharing is performed such that register **p** is used to store both **t1** and **t2**, whereas register **q** is used to store **t3**, **t4**, and **t5**. Result **t6** has a dedicated register.

a) Define the term "resource dominated", and state whether this circuit is resource dominated.

[2]

b) Figure 5.1 shows a partially completed ROM table for a horizontal microcode implementation. State the value of N, the final address, and complete the table in binary.

[5]

Address	Select lines at input to (+,1)	Select lines at input to (*,1)	Select lines at input to register **p**	Select lines at input to register **q**	Enable line for register **p**	Enable line for register **q**	Enable line for register **t6**
0							
...							
N							

Figure 5.1

c) Remove any duplicate control lines (those for which there is another control line with which they have equal value at all time instants) and superfluous control lines (those that have the same value for all time), to obtain a more compact horizontal microcode representation. Explain the steps in your answer.

[3]

d) Re-design your controller using the principle of distributed control, stating the contents of each ROM. You may use horizontal microcode. State clearly any reasons behind design choices you have made.

[7]

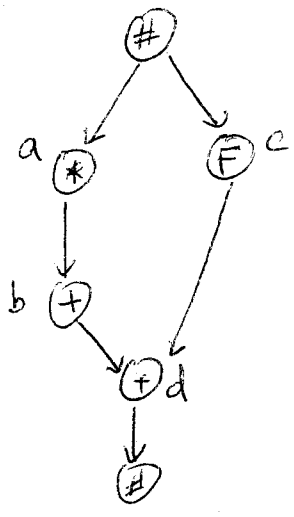
e) State Rent's rule. Given that the circuit has gate-level Rent constants $\beta = 0.5$, $K = 2.0$, estimate the number of gates in the circuit.

[3]

Synthesis of Digital Architectures Solutions 2005

E4.43
ISE 4.43
A01

a)



b)

v	S(v)
a	0
b	2
c	0
d	3

Overall latency = 4 cycles

c) $Y(a) = (*, 1)$
$Y(b) = (+, 1)$
$Y(c) = (ROM, 1)$
$Y(d) = (+, 1)$

Assuming resource-dominated, area = 1 + 2 + 10 = 13.

d) Need to get it range $[-1, 1]$

$c \in [-1, 0] \Rightarrow 2c+1 \in [-1, 1]$

Let $x = 2c + 1$

We want to express $f(x) = \exp\left(\frac{1}{2}(x-1)\right)$ as a weighted sum of legendre polynomials.

$\phi_0(x) = 1$

$\phi_1(x) = \frac{1}{2}\frac{d}{dx}\{x^2 - 1\}$

$= x$

$\phi_2(x) = \frac{1}{(4)(2)}\frac{d^2}{dx^2}\left\{(x^2-1)^2\right\}$

$= \frac{1}{8}\frac{d^2}{dx^2}\left\{x^4 - 2x^2 + 1\right\}$

$= \frac{1}{8}\frac{d}{dx}\left\{4x^3 - 4x\right\}$

$= \frac{1}{2}(3x^2 - 1)$

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

$\langle \phi_1, \phi_1 \rangle = \int_{-1}^{1} x^2\,dx = \left[\frac{1}{3}x^3\right]_{-1}^{1} = 2/3$

$$\langle \phi_2, \phi_2 \rangle = \frac{1}{4}\int_{-1}^{1}(3x^2-1)^2\,dx$$

$$= \frac{1}{4}\int_{-1}^{1}(9x^4-6x^2+1)\,dx = \frac{1}{4}\left[\frac{9}{5}x^5 - 2x^3 + x\right]_{-1}^{1}$$

$$= \frac{1}{2}\left[\frac{9}{5} - 2 + 1\right]$$

$$= \underline{2/5}$$

$$\langle f, \phi_0 \rangle = \int_{-1}^{1} e^{\frac{1}{2}(x-1)}\,dx = e^{-\frac{1}{2}}\int_{-1}^{1} e^{+\frac{1}{2}x}\,dx$$

$$= 2e^{-\frac{1}{2}}\left[e^{\frac{1}{2}x}\right]_{-1}^{1} = 2e^{-\frac{1}{2}}\left(e^{\frac{1}{2}} - e^{-\frac{1}{2}}\right)$$

$$= 2\left(1 - e^{-1}\right)$$

$$\langle f, \phi_1 \rangle = \int_{-1}^{1} x\, e^{\frac{1}{2}(x-1)}\,dx = \left[2x\, e^{\frac{1}{2}(x-1)}\right]_{-1}^{1} - 2\int_{-1}^{1} e^{\frac{1}{2}(x-1)}\,dx$$

(x = v, $e^{\frac{1}{2}(x-1)}$ = $\frac{du}{dx}$)

$$\frac{dv}{dx} = 1 \quad u = 2e^{\frac{1}{2}(x-1)}$$

$$= 2 + 2e^{-1} - 4 + 4e^{-1}$$

$$= 2\left(3e^{-1} - 1\right)$$

$$\langle f, \phi_2 \rangle = \frac{1}{2}\int_{-1}^{1}(3x^2-1)\,e^{\frac{1}{2}(x-1)}\,dx = \frac{3}{2}\int_{-1}^{1} x^2\, e^{\frac{1}{2}(x-1)}\,dx$$

(x^2 = v, $e^{\frac{1}{2}(x-1)}$ = u)

$$-\{ \quad -1 + e^{-1}$$

$$\frac{dv}{dx} = 2x \qquad u = 2e^{\frac{1}{2}(x-1)}$$

$$\langle f, \varphi_2 \rangle = \frac{3}{2}\left[2x^2 e^{\frac{1}{2}(x-1)}\right]_{-1}^{1} - \frac{3}{2}\int_{-1}^{1} 4x\, e^{\frac{1}{2}(x-1)}\,dx$$

$$-1+e^{-1}$$

$$= 3\left(1 - e^{-1}\right) - (6)(2)\left(3e^{-1} - 1\right)$$

$$-1+e^{-1}$$

$$= 3 - 3e^{-1} - 36e^{-1} + 12 - 1 + e^{-1}$$
$$= 14 - 38e^{-1}$$
$$= 2(7 - 19e^{-1})$$

So $g(x) = \frac{2(1-e^{-1})}{2} + \frac{(3)(2)(3e^{-1}-1)}{2}x$

$$+ \frac{(5)(2)(7-19e^{-1})}{2}\cdot\frac{1}{2}(3x^2-1)$$

$$= (1-e^{-1}) + 3(3e^{-1}-1)x + \frac{5}{2}(7-19e^{-1})(3x^2-1)$$

So $a_0 = 1 - e^{-1} - \frac{5}{2}(7-19e^{-1}) = \frac{1}{2}(2 - 2e^{-1} - 35 + 95e^{-1})$

$a_1 = 3(3e^{-1}-1)$ $= \frac{1}{2}(-33 + 93e^{-1})$

$a_2 = \frac{15}{2}(7-19e^{-1})$

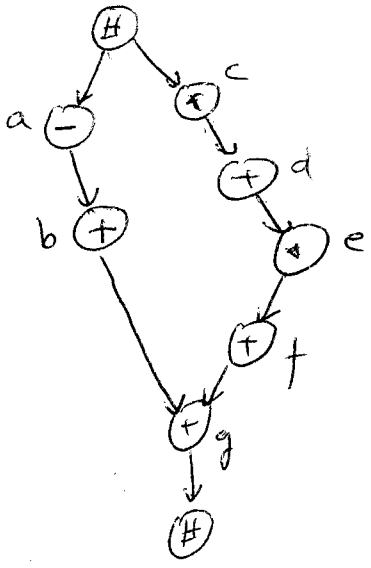
e)

$63*b = 64*b - b$

$= (b << 6) - b$

(c free)

& $g(x) = a_0 + x(a_1 + a_2 x)$



f)

v	s(v)
a	0
b	1
c	0
d	2
e	3
f	5
g	6

Clearly no * conflicts
Also no + conflicts

Area = 1+2 = 3, under the assumption of resource domination.

2. Let $W = \bigcup_{v \in V} \{ASAP_v, \ldots, ALAP_v\}$

Then

$$\text{Minimize}: \sum_{r \in R} c_r \sum_{i=1}^{a_r} b_{ir}$$

subject to:

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

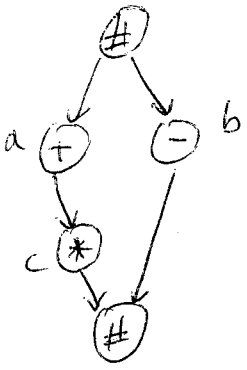
$$\forall t \in W, \forall r \in R, \forall i \in \{1, \ldots, a_r\},$$

$$\sum_{v \in V : r \in T(v)} \sum_{t' \in \{t, \ldots, t+d_r-1\} \cap \{ASAP_v, \ldots, ALAP_v - d_r + d_{min\,v}\}} x_{vt'ir} \leq b_{ir} \qquad (2)$$

$$\forall (v', v) \in E$$

$$\sum_{r \in T(v)} \sum_{i=1}^{a_r} \sum_{t=ASAP_v}^{ALAP_v - d_r + d_{min\,v}} t \, x_{vtir} \geq \sum_{r \in T(v')} \sum_{i=1}^{a_r} \sum_{t=ASAP_{v'}}^{ALAP_{v'} - d_r + d_{min\,v'}} (t + d_r) x_{vtir} \qquad (3)$$

b)



Objective $\sum_{r \in \{mult, ripple, lookahead\}} c_r \sum_{i=1}^{a_r} b_{i,r}$

$= c_{mult}\, b_{1,mult} + c_{ripple}\, b_{1,ripple} + c_{lookahead}\, b_{1,lookahead}$

$= 4 b_{1,mult} + 2 b_{1,ripple} + 3 b_{1,lookahead}$

Constraint set (1):

$$\forall v \in \{a,b,c\} \sum_{r \in T(v)} \sum_{i=1}^{a_r} \sum_{t=ASAP_v}^{ALAP_v - d_r + d_{min\,v}} x_{vt,ir} = 1$$

Need ASAP & ALAP:

v	ASAP(v)	ALAP(v)
a	0	0
b	0	1
c	1	1

So (1) becomes

(a) $$\sum_{r \in \{\text{ripple, lookahead}\}} \sum_{t=0}^{0-d_r+1} x_{a,t,1,r} = 1$$

i.e. $$x_{a,0,1,\text{lookahead}} = 1 \qquad (A)$$

(b) $$\sum_{r \in \{\text{ripple, lookahead}\}} \sum_{t=0}^{1-d_r+1} x_{b,t,1,r} = 1$$

i.e. $$x_{b,0,1,\text{lookahead}} + x_{b,1,1,\text{lookahead}} + x_{b,0,1,\text{ripple}} = 1 \qquad (C)$$

(c) $$\sum_{t=1}^{1} x_{c,t,1,\text{mult}} = 1$$

i.e. $$x_{c,1,1,\text{mult}} = 1 \qquad (B)$$

Ad (2) becomes

$\forall t \in \{0,1\}$, $\forall r \in \{\text{ripple, lookahead, mult}\}$, s

$$\sum_{v \in \{a,b,c\} : r \in T(v)} \sum_{t' \in \{t, \ldots, t+d_r-1\} \cap \{ASAP_v, \ldots, ALAP_v - d_r + 1\}} x_{v,t',1,r} \leq b_{1,r}$$

$$\forall \sum_{v \in \{a,b,c\} : r \in T(v)} x_{v,t}$$

i.e. the following pairs have non-empty time intersections
$(t=0,\ r=\text{ripple})$

$$\sum_{v\in\{a,b\}} \sum_{t\in\{0,1\}\cap\{ASAP_v,\ldots,ALAP_v-1\}} x_{v,t,1,\text{ripple}} \le b_{1,\text{ripple}}$$

$$\Rightarrow x_{b,0,1,\text{ripple}} \le b_{1,\text{ripple}} \qquad (c)$$

$(t=0,\ r=\text{lookahead})$

$$\sum_{v\in\{a,b\}} x_{v,0,1,\text{lookahead}} \le b_{1,\text{lookahead}}$$

$$\Rightarrow x_{a,0,1,\text{lookahead}} + x_{b,0,1,\text{lookahead}} \le b_{1,\text{lookahead}} \qquad (d)$$

$(t=1,\ r=\text{lookahead})$

$$\sum_{v\in\{a,b\}} \sum_{t'\in\{1\}\cap\{ASAP_v,\ldots,ALAP_v\}} x_{v,t',1,\text{lookahead}} \le b_{1,\text{lookahead}}$$

$$\Rightarrow x_{b,1,1,\text{lookahead}} \le b_{1,\text{lookahead}} \qquad (e)$$

$(t=1,\ r=\text{mult})$

$$x_{c,1,1,\text{mult}} \le b_{1,\text{mult}} \qquad (f)$$

We have $x_{a,0,1,\text{locked}} = 1$ & $x_{c,1,1,\text{built}} = 1$

$\Rightarrow b_{1,\text{locked}} = 1$ (D) $\Rightarrow b_{1,\text{built}} = 1$ (F)

Variables with currently are

(V1) $b_{1,\text{ripple}}$; (V2) $x_{b,0,1,\text{ripple}}$; (V3) $x_{b,0,1,\text{locked}}$; (V4) $x_{b1,1,\text{locked}}$

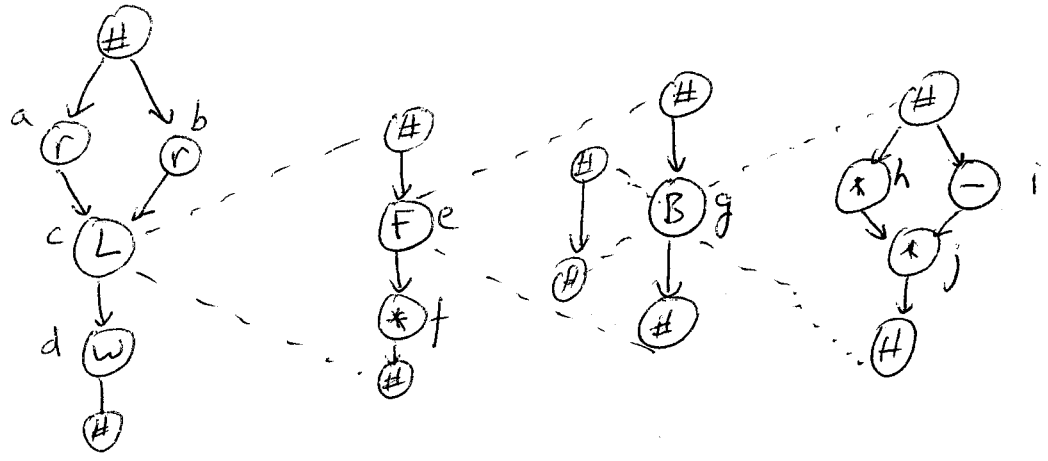
V1	V2	V3	V4	CONSTRAINT C	D	G	FEASIBLE?	OBJECTIVE $7+2b_{1,\text{ripple}}$
0	0	0	0	✓	✓	X	X	
0	0	0	1	✓	✓	✓	✓	7
0	0	1	0	✓	X	✓	X	
0	0	1	1	✓	X	X	X	
0	1	0	0	X	✓	✓	X	
0	1	0	1	X	✓	X	X	
0	1	1	0	X	X	X	X	
0	1	1	1	X	X	X	X	
1	0	0	0	✓	✓	X	X	
1	0	0	1	✓	✓	✓	✓	9
1	0	1	0	✓	X	✓	X	
1	0	1	1	✓	X	X	X	
1	1	0	0	✓	✓	✓	✓	9
1	1	0	1	✓	✓	X	X	
1	1	1	0	✓	X	X	X	
1	1	1	1	✓	X	X	X	

Constrs A, B, E, F are always satisfied.

Optimum soln is with $b_{1,\text{ripple}} = 0$, $x_{b,0,1,\text{ripple}} = 0$, $x_{b,0,1,\text{locked}} = 0$, $x_{b,1,1,\text{locked}} = 1$.

Optimum objective $= \underline{7}$.

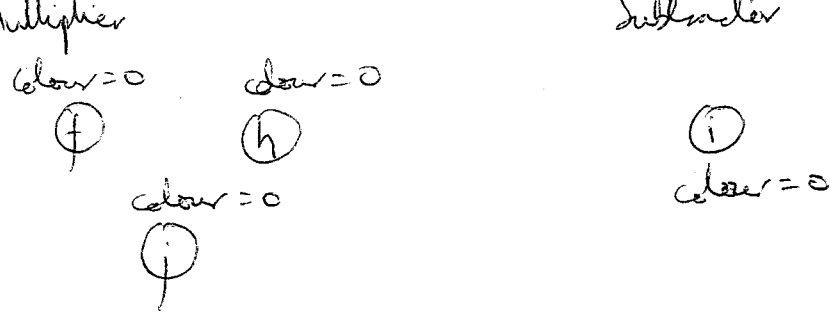
3 a)



b)

v	s(v)
a	0
b	0
c	0
d	6N
e	0 + 6(i - 1)
f	4 + 6(i - 1)
g	0 + 6(i - 1)
h	0 + 6(i - 1)
i	0 + 6(i - 1)
j	2 + 6(i - 1)

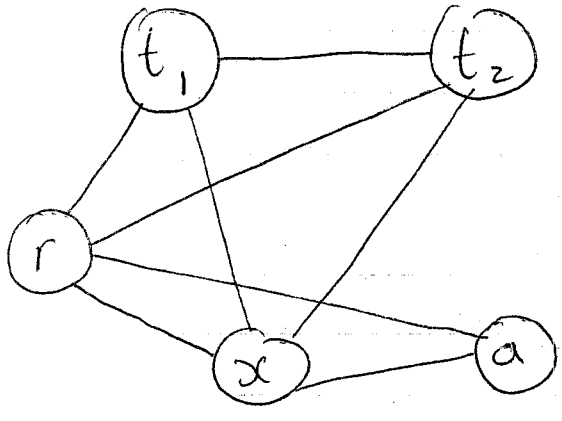
c)



d) Given p the label t_1 & $1-p$ the label t_2,
we have variables

t_1 ; t_2 ; r ; x ; a

Variable	Cycle(s) Produced	Cycle(s) Consumed	Register Needed
t_1	$6(i-1)+1$	$6(i-1)+2$	$[6(i-1)+2, 6(i-1)+4]$
t_2	$6(i-1)$	$6(i-1)+2$	$[6(i-1)+1, 6(i-1)+2]$
r	0	$6(i-1)$	$[0, 6(N-1)]$
x	0	$6(i-1)$	$[1, 6(N-1)]$
a	$6(i-1)+3$	$6(i-1)+4$	$[6(i+1)+4, 6(i-1)+4]$



It is an interval graph, e.g.

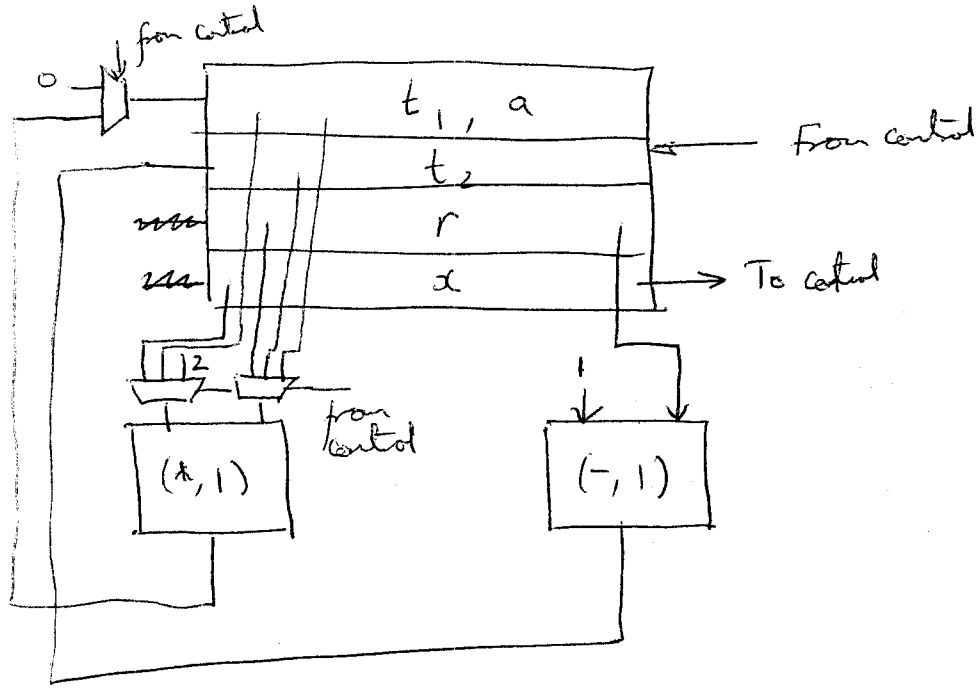
$r \Leftrightarrow [0, 10]$
$x \Leftrightarrow [0, 10]$
$t_1 \Leftrightarrow [0, 1]$
$t_2 \Leftrightarrow [0, 1]$
$a \Leftrightarrow [2, 3]$

Coloring

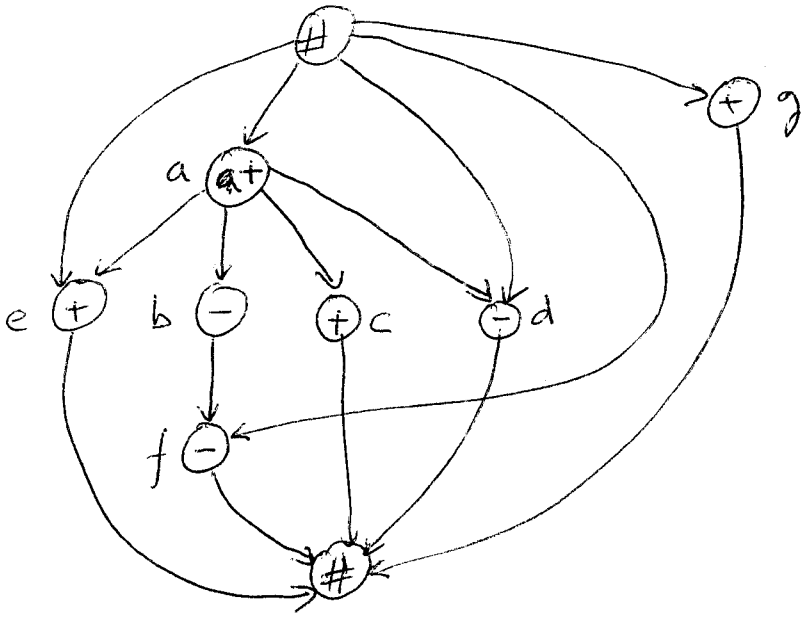
t_1	0
t_2	1
r	2
x	3
a	0

f) There is a 4-clique $\{t_1, t_2, r, x\}$ & $\chi(G) \geq \kappa(G) = 4$.

g)



4 a)



b) 3 cycles (a, b, f)

c) Resource - constrained list - scheduling

d) Call bound B.

~~For b=~~

First, calculate (min)ALAP times for urgency:

v	S(v)
a	0
b	1
c	2
d	2
e	2
f	2
g	2

For B = 1

S(a) = 0
S(b) = 1
S(c) = 2
S(d) = 3
S(e) = 4
S(f) = 5
S(g) = 6

For B = 2

S(a) = 0
S(g) = 0
S(b) = 1
S(c) = 1
S(d) = 2
S(e) = 2
S(f) = 3

For B = 3

S(a) = 0
S(g) = 0
S(b) = 1
S(c) = 1
S(d) = 1
S(e) = 2
S(f) = 2

For B = 4

S(a) = 0
S(g) = 0
S(b) = 1
S(c) = 1
S(d) = 1
S(e) = 1
S(f) = 2

For B > 4, result will be same as for B = 4.

c) Latency-constrained list-scheduling

+)

$\lambda = 3$

$s(a) = 0$
$s(b) = 1$
$s(c) = s(d) = s(e) = s(f) = s(g) = 2$

$\lambda = 4$
$s(a) = 0$
$s(b) = 1$
$s(c) = 2$
$s(d) = s(e) = s(f) = s(g) = 3$

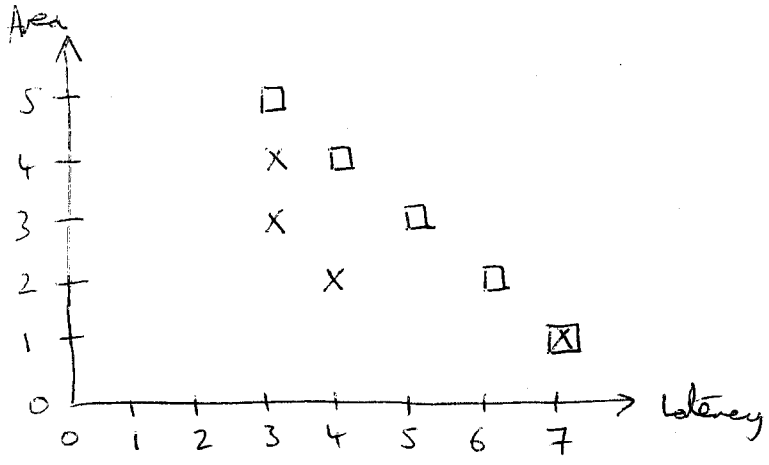
$\lambda = 5$
$s(a) = 0$
$s(b) = 1$
$s(c) = 2$
$s(d) = 3$
$s(e) = s(f) = s(g) = 4$

$\lambda = 6$
$s(a) = 0$
$s(b) = 1$
$s(c) = 2$
$s(d) = 3$
$s(e) = 4$
$s(f) = s(g) = 5$

$\lambda = 7$
$s(a) = 0$
$s(b) = 1$
$s(c) = 2$
$s(d) = 3$
$s(e) = 4$
$s(f) = 5$
$s(g) = 6$

For $\lambda > 7$, result will be as for $\lambda = 7$.

g)



h) An inferior design is one that is worse than other designs in at least one objective, and no better in the remaining objectives.

i) Yes LC-list-scheduling because it puts off allocation of resources until it is forced to by ALAP times, resulting in more resources in the later time slots.

j) A Pareto-optimal design is one that cannot be improved in one objective without sacrificing another objective.

k) No. Both algorithms are poly-time, but scheduling is NP-hard under constraints.

5 a) A resource dominated circuit is one whose computational resources dominate the calculation of area, power, speed, etc. This is not resource dominated.

b) N = 2

Address	(+,1)	(*,1)	Sel p	Sel q	En p	En q	En t6
0	0	00	0	00	1	1	0
1	1	01	1	01	1	1	0
2	X	10	X	10	0	1	1
	↑(1)	↑(5) ↑(2)	↑(3)	↑(6) ↑(4)		↑(8)	↑(7)

c) Lines labelled (1-4) above can be made identical. (A)
Lines labelled (5-7) are identical. (B)
Line (8) is superfluous

Address	A	B	En p
0	0	0	1
1	1	0	1
2	0	1	0

d) One possible idea is to group by functional unit.

Group 1

Address	(+,1)	En p	En q	Sel p	Sel q
0	0	1	1	0	00
1	1	1	1	1	01
2	X	0	1	X	10

Group 2

Address	(x, 1)	$t_i^{G_1}$
0	00	0
1	01	0
2	10	1

[others already generated in Group 1]

e) $N = K G^{\beta}$

where N = No. pins

G = No. gates

K, β : Rent constants

This circuit has 16 input bits & 48 output bits

$$N = 16 + 48 = 64$$

$$64 = 2G^{1/2}$$

$$\Rightarrow G = 32^2$$

$$G = 1024$$

So ~1024 Gates