E4.43
AO1
ISE4.43

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2004

MSc and EEE/ISE PART IV: MEng and ACGI

SYNTHESIS OF DIGITAL ARCHITECTURES

Monday, 17 May 10:00 am

Time allowed: 3:00 hours

There are FIVE questions on this paper.

Answer THREE questions.

Corrected Copy

All questions carry equal marks

Q3 c)

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

Examiners responsible First Marker(s) : G.A. Constantinides
Second Marker(s) : P.Y.K. Cheung

1.

In numerical analysis, a common task is to estimate the value of the differential of a function. Some pseudo-code for this task is shown below.

```
diff( x, h, f0, f1, f2 ) {
  p = x/h;
  t = (p-0.5)*f0 - 2*p*f1 + (p+0.5)*f2;
  return (t/h);
}
```

This behaviour is to be implemented by an architecture in which there are three types of resource: adder/subtractor (latency 1 cycle); multiplier (latency 2 cycles); divider (latency 2 cycles).

a) Construct a Control Data Flow Graph for the algorithm `diff`.

[2]

b) Perform an ASAP and an ALAP scheduling on your CDFG for the lowest possible latency constraint, stating the ASAP time, ALAP time, and mobility of each node.

[4]

c) State the minimum overall latency of this design.

[1]

You wish to schedule this behaviour given that at most 1 multiplier, 1 divider, and 4 adder/subtractors can be used.

d) Name an appropriate algorithm to solve this problem.

[1]

e) Solve this scheduling problem, stating the scheduled start time of each node and the overall latency required.

[6]

To find the optimal solution to a general scheduling problem for a DFG $G(V,E)$, it is proposed to use Integer Linear Programming. You are given a set of binary variables x_{vt} for each $v \in V$ and time-step t, equal to 1 iff operation v is scheduled at time step t; a latency (delay) d_v for each node $v \in V$; a function $T : V \rightarrow R$, and an upper-bound a_r on each resource type $r \in R$. You also know the ASAP and ALAP times ASAP_v and ALAP_v for each node $v \in V$, and the overall latency constraint λ.

f) Formulate the constraint "each operation must have a unique start time" as a set of linear constraints in x_{vt}.

[2]

g) Formulate the data dependency constraints as a set of linear constraints in x_{vt}.

[2]

h) Formulate the constraint "no more than a_r resources of type r can be used" as a set of linear constraints in x_{vt}.

[2]

2.

Some pseudo-code for a main program `main()`, and two functions `sin()` and `cos()`, is shown below.

```
main() {
   read y;
   x = 2*y;
   for i = 1 to 2 {
        read y;
        c = y > 0;
        if( c ) {
             x = x + sin( y );
        } else {
             x = x + cos( y );
        }
   }
}

sin( x ) {
   return ( x - (0.167*x)*(x*x) );
}

cos( x ) {
   return ( 1 - 0.5*x*x );
}
```

a) Create a CDFG for the above code, representing `read` calls as `read` nodes, and give each node a unique label.

[6]

You have a library of two resource types: ALU (capable of performing addition, subtraction, and comparison); and multiplier (capable of performing multiplication).

b) Derive an ASAP schedule for this code, under the assumption that all operations, reads and writes take one cycle to execute, while "start task" and "end task" nodes take zero cycles. For each labelled node, state the ASAP schedule time(s), noting that some nodes will be scheduled in more than one cycle.

[6]

[Question 2 continues on page 3]

[Continuation of Question 2]

It is suggested that the performance of the design may be improved by repeating the body of the loop, instead of using a loop construct.

The equivalent pseudo-code is shown below.

```
main() {
   read y;
   x = 2*y;
   read y1;
   c1 = y1 > 0;
   if( c1 ) {
        x1 = x + sin( y1 );
   } else {
        x1 = x + cos( y1 );
   }
   read y2;
   c2 = y2 > 0;
   if( c2 ) {
        x2 = x1 + sin( y2 );
   } else {
        x2 = x1 + cos( y2 );
   }
}

sin( x ) {
   return ( x - (0.167*x)*(x*x) );
}

cos( x ) {
   return ( 1 - 0.5*x*x );
}
```

c) Derive the minimum overall latency for the modified code, and account for any difference in performance.

[8]

3.

a) Describe the initialisation problem in retiming. [2]

Some pseudo-code is shown below.

```
bq1 = 0; bq2 = 0; cq = 0;
while( true ) {
    read x;
    a = 2*x;
    c = a + bq2;
    b = a*c;
    write cq;
    bq2 = bq1; bq1 = b; cq = c;
}
```

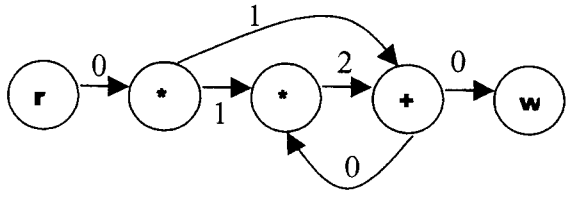
b) Construct a delay-weighted DFG to represent the inner loop of this code, using nodes of type "*", "+", "r" (read) and "w" (write). [6]

c) Given the symbols shown in Figure 1, formulate the retiming problem for a general delay-weighted DFG $G(V,E)$ as an integer linear program. [6]

L	- length of the longest combinational path in the DFG
s_v	- length of the longest combinational path up to the input of node v
$d(v)$	- the combinational delay of node v
$x_{u,v}$	- binary decision variable. $x_{u,v} = 1$ iff $w_r(u,v) = 0$ (equivalently $x_{u,v} = 0$ iff $w_r(u,v) \geq 0$)
N	- large negative number
P	- large positive number
$w_r(u,v)$	– retimed weight of edge (u,v)
$w(u,v)$	- original weight of edge (u,v)
$r(v)$	- retiming value of node v

Figure 1

After retiming the code shown above, the delay-weighted DFG shown below is obtained.



d) Derive the pseudo-code corresponding to the retimed DFG. [6]

4.

A polynomial $g(x)$ of order n can be expressed as a linear combination of basis functions $\{\phi_i(x)\}$, where ϕ_i is a polynomial of order i (Equation 1).

$$g(x) = \sum_{i=0}^{n} a_i \phi_i(x) \qquad \text{(Equation 1)}$$

a) Given the inner-product definition of (Equation 2), prove that (Equation 3) defines the least-squares approximation to $f(x)$ under the condition that ϕ_i and ϕ_j are orthogonal for $i \neq j$.

$$< f, g >= \int_{-1}^{1} f(x)g(x)dx \qquad \text{(Equation 2)}$$

$$a_i = \frac{< f, \phi_i >}{< \phi_i, \phi_i >} \qquad \text{(Equation 3)}$$

[6]

The Legendre polynomials are defined by (Equation 4).

$$\phi_i(x) = \frac{1}{2^i i!} \frac{d^i}{dx^i} \left\{ (x^2 - 1)^i \right\} \qquad \text{(Equation 4)}$$

b) Derive a first order least-squares polynomial approximation to the $\sin(\pi x)$ function for x over [-1,1] as a weighted sum of Legendre polynomials.

[4]

c) Construct the CDFG corresponding to the Horner's scheme evaluation of a 3rd order polynomial.

[2]

The polynomial is to be evaluated using multiplier and adder resources, each of which has unit latency.

d) Schedule this Horner's scheme evaluation (i) with ASAP scheduling, and (ii) under the resource constraints of one multiplier and one adder. Provide the start time for each operation.

[4]

An alternative scheme for evaluating a 3rd order polynomial is to use the code $y = (c_0 + c_1 * x) + (x * x) * (c_2 + c_3 * x)$.

e) Schedule the evaluation under this alternative scheme (i) with ASAP scheduling and (ii) under the resource constraints of one multiplier and one adder. Provide the start time for each operation.

[4]

5.

Some pseudo-code for a memory-mapped co-processor responsible for calculating a quadratic function is shown below. `read(x)` corresponds to a read of the address bus, and `write(y)` corresponds to a write to the data bus. `signal_done()` corresponds to asserting an external "done" signal.

```
read(x);
y = x*x + 3*x + 5;
signal_done();
write(y);
```

This behaviour is to be implemented using multiplier resources of latency 2 cycles and adder resources of latency 1 cycle. `read`, `write`, and `signal_done` have zero latency.

a) Construct a control/data flow graph (CDFG) for this code, including nodes for "read" and "write", and "signal_done".

[5]

b) Perform an ASAP scheduling on this CDFG.

[4]

The bus specification requires data to be written to the data bus at least two cycles and at most five cycles after the address is read. In addition it is required that the "done" signal must be asserted exactly one clock cycle before the output is written to the data bus.

c) Construct an edge-weighted graph corresponding to this computation, such that the longest-path from the source node to each other node forms the ASAP time of the latter node under the above timing constraints.

[4]

d) Perform an ASAP scheduling under these timing constraints.

[4]

Your project manager asks you to additionally ensure that the "done" signal is produced no later than the first addition is calculated.

e) Re-draw the edge-weighted graph from part (c) to incorporate this constraint.

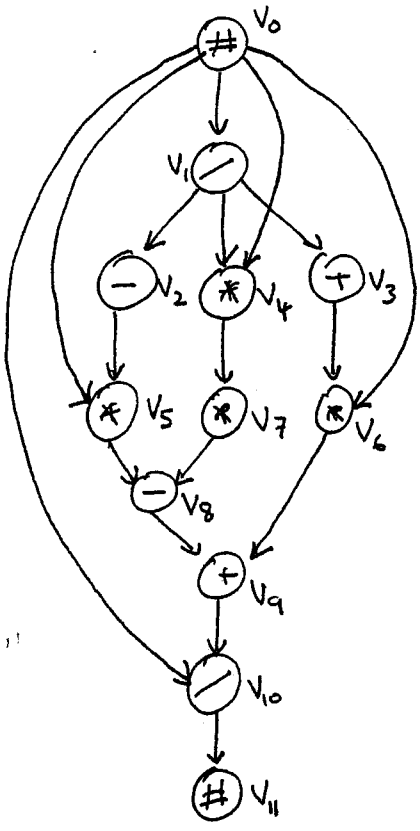
[1]

f) By reference to a cycle in the graph, demonstrate that your manager is asking the impossible.

[2]

SYNTHESIS OF DIGITAL ARCHITECTURES
MODEL SOLUTIONS

1. a)



[2 MARKS]

b)

NODE	ASAP	ALAP	MOBILITY
V_0	0	0	0
V_1	0	0	0
V_2	2	3	1
V_3	2	4	2
V_4	2	2	0
V_5	3	4	1
V_6	3	5	2
V_7	4	4	0
V_8	6	6	0
V_9	7	7	0
V_{10}	8	8	0
V_{11}	10	10	0

[4 MARKS]

c) 10 CYCLES [1 MARK]

d) RESOURCE-CONSTRAINED LIST SCHEDULING [1 MARK]

1. e)

CYCLE	READY LIST	NODES SCHEDULED
0	$\{v_1\}$	$\{v_1\}$
1	$\emptyset$	$\emptyset$
2	$\{v_2, v_3, v_4\}$	$\{v_2, v_3, v_4\}$
3	$\{v_5, v_6\}$	$\emptyset$
4	$\{v_5, v_6, v_7\}$	$\{v_5\}$ [or $\{v_7\}$]
5	$\{v_6, v_7\}$	$\emptyset$
6	$\{v_6, v_7\}$	$\{v_7\}$
7	$\{v_6\}$	$\emptyset$
8	$\{v_6, v_8\}$	$\{v_6, v_8\}$
9	$\emptyset$	$\emptyset$
10	$\{v_9\}$	$\{v_9\}$
11	$\{v_{10}\}$	$\{v_{10}\}$
12	$\emptyset$	$\emptyset$
13	$\{v_{11}\}$	$\{v_{11}\}$

OVERALL LATENCY = 13 CYCLES [6 MARKS]

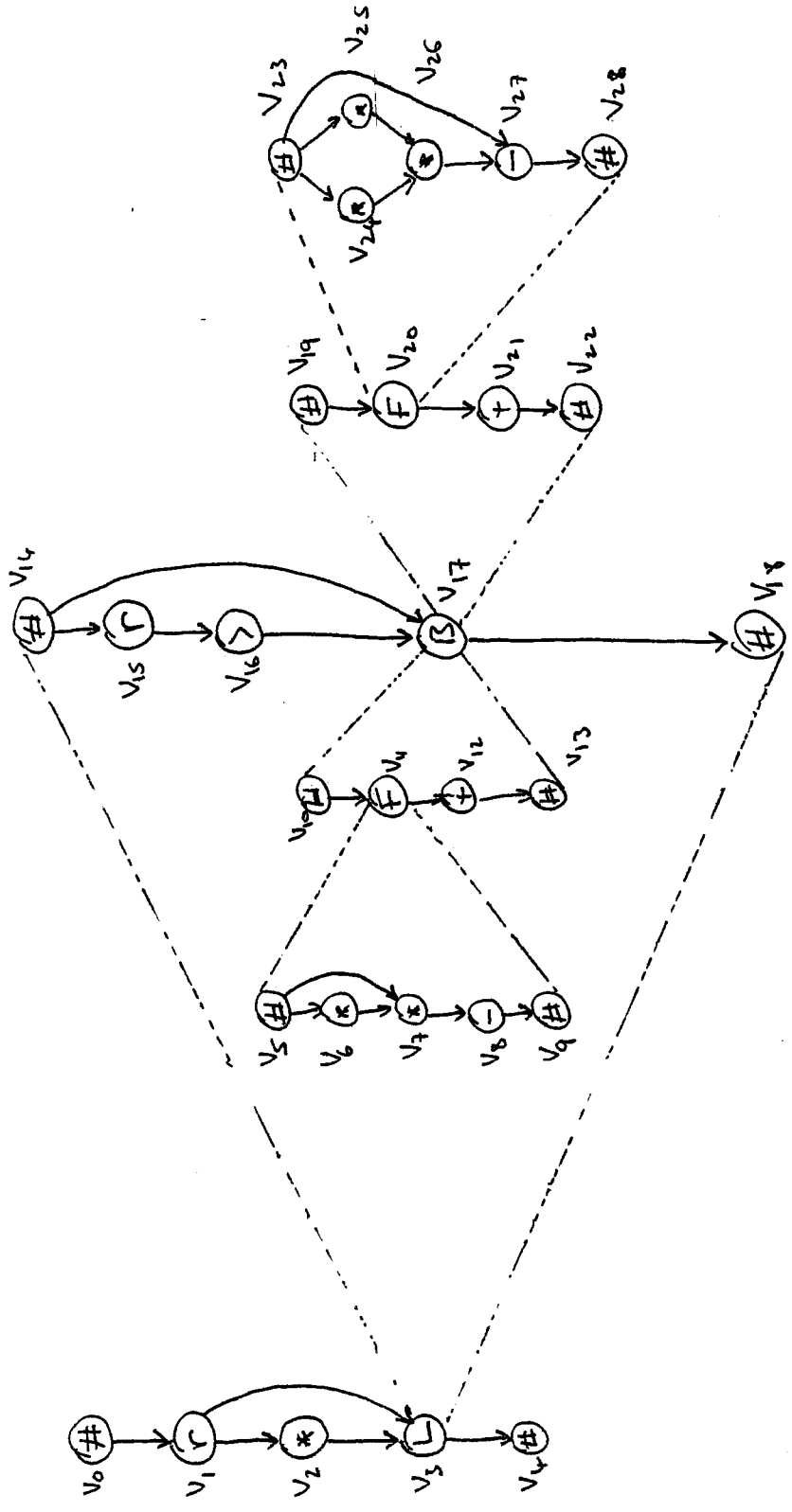
f) $\forall v \in V, \sum_{t=ASAP_v}^{ALAP_v} x_{vt} = 1$ [2 MARKS]

g) $\forall (v_1, v_2) \in E, \sum_{t=ASAP_{v_1}}^{ALAP_{v_1}} t \cdot x_{v_1 t} + d_{v_1} \leq \sum_{t=ASAP_{v_2}}^{ALAP_{v_2}} t \cdot x_{v_2 t}$

h) $\forall t \in \{0, \ldots, \lambda\}, \forall r \in R,$

$$\sum_{v \in V : T(v) = r} \sum_{t' \in \{t - d_v + 1, \ldots, t\} \cap \{ASAP_v, \ldots, ALAP_v\}} x_{vt'} \leq a_r$$

2. a)


V0
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
V17
V18
V19
V20
V21
V22
V23
V24
V25
V26
V27
V28

[6 MARKS]

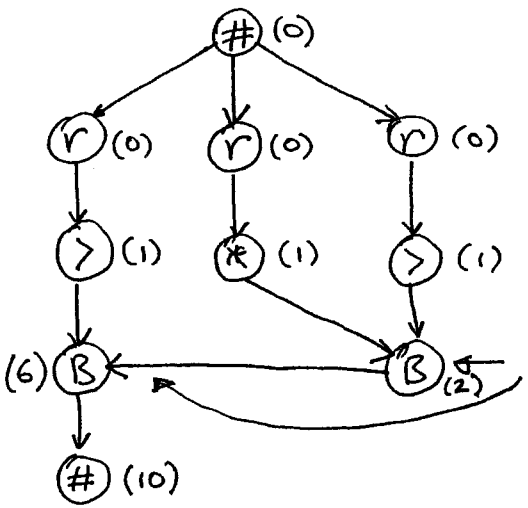
2. b)

NODE	ASAP TIME (s)
V_0	0
V_1	0
V_2	1
V_3	2
V_4	14
V_5	4, 10
V_6	4, 10
V_7	5, 11
V_8	6, 12
V_9	7, 13
V_{10}	4, 10
V_{11}	4, 10
V_{12}	7, 13
V_{13}	8, 14
V_{14}	2, 8
V_{15}	2, 8
V_{16}	3, 9
V_{17}	4, 10
V_{18}	8, 14
V_{19}	4, 10
V_{20}	4, 10
V_{21}	7, 11
V_{22}	8, 14
V_{23}	4, 10
V_{24}	4, 10
V_{25}	4, 10
V_{26}	5, 11
V_{27}	6, 12
V_{28}	7, 13

[6 MARKS]

2. (c)

ASAP TIMES IN PARENTHESES



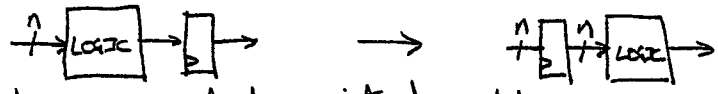
ALTERNATIVE BRANCHES AS PER PART (a)

OVERALL LATENCY = 10 CYCLES

DIFFERENCES DUE TO INCREASED SCOPE FOR PARALLELISM DUE TO CODE MOTION OUT OF LOOP.

[8 MARKS]

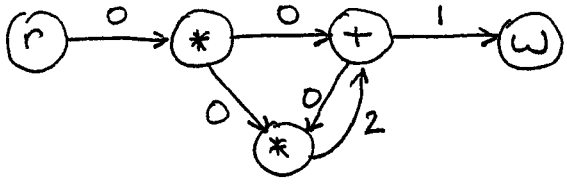
3. (a) If we wish to retime backwards, e.g.



How do we select initial state of registers to maintain behaviour?

[2 MARKS]

(b)



[6 MARKS]

(c) MINIMIZE L SUBJECT TO

$$s_v \geq s_u + d(u) + x_{uv} N \quad \text{for all } (u,v) \in E$$

$$w_r(u,v) \leq x_{u,v} P \quad \text{for all } (u,v) \in E$$

$$s_v + d(v) \leq L \quad \text{for all } v \in V$$

$$w_r(u,v) = w(u,v) + r(v) - r(u) \quad \text{for all } (u,v) \in E$$

$$w_r(u,v) \geq 0 \quad \text{for all } (u,v) \in E$$

$$x_{u,v} \in \{0,1\}$$

[6 MARKS]

(d)

```
aq = 0;   bq2 = 0;   bq1 = 0;
while(TRUE) {
        read  x;
        a = 2 * x;
        c = aq + bq2;
        b = aq + c;
        write c;
        bq2 = bq1;   bq1 = b;   aq = a;
}
```

4. (a) Let ϵ = Error.

Then $$\epsilon = \int_{-1}^{1} \left(f(x) - g(x)\right)^2 dx$$

$$= \int_{-1}^{1} \left\{ f(x) - \sum_{i=0}^{n} a_i \phi_i(x) \right\}^2 dx$$

$$= \int_{-1}^{1} f^2(x) - 2 \sum_{i=0}^{n} a_i \int_{-1}^{1} f(x)\phi_i(x)\,dx + \sum_{i=0}^{n} \sum_{j=0}^{n} a_i a_j \int_{-1}^{1} \phi_i(x)\phi_j(x)\,dx$$

$$= \int_{-1}^{1} f^2(x) - 2 \sum_{i=0}^{n} a_i \langle f, \phi_i \rangle + \sum_{i=0}^{n} a_i^2 \langle \phi_i, \phi_i \rangle$$

$$\frac{\partial \epsilon}{\partial a_i} = -2\langle f, \phi_i \rangle + 2a_i \langle \phi_i, \phi_i \rangle$$

Set $\frac{\partial \epsilon}{\partial a_i} = 0$. Then

$$a_i = \frac{\langle f, \phi_i \rangle}{\langle \phi_i, \phi_i \rangle}$$

[6 MARKS]

(b) $\phi_0(x) = 1$

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

$$= \tfrac{1}{2}(2x) = x$$

$$a_0 = \int_{-1}^{1} \sin \pi x\, dx = -\frac{1}{\pi}\left[\cos \pi x\right]_{-1}^{1} = -\frac{1}{\pi}\left\{(-1) - (-1)\right\} = 0$$

$$a_1 = \int_{-1}^{1} x \sin \pi x\, dx = \left[-\frac{x}{\pi}\cos \pi x\right]_{-1}^{1} + \frac{1}{\pi}\int_{-1}^{1} \cos \pi x\, dx$$

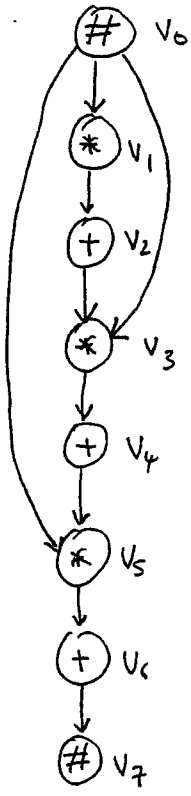
$$= -\frac{1}{\pi}\left[-1 - 1\right] - \frac{1}{\pi^2}\left[\sin \pi x\right]_{-1}^{1}$$

$$= \frac{2}{\pi}$$

So $g(x) \approx \frac{2}{\pi} \sin \pi x$ with $g(x) = \frac{2}{\pi} x$.

[4 MARKS]

4. (c) $y = c_0 + x\left[c_1 + x\left[c_2 + x c_3\right]\right]$

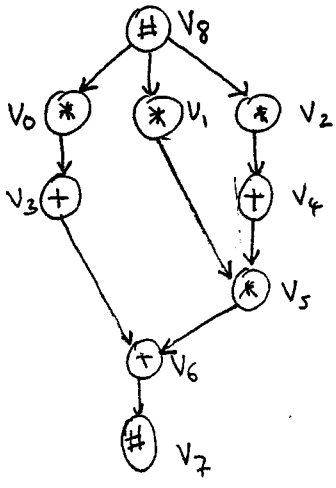


[2 MARKS]

(a)

NODE v	ASAP S(v)	RESOURCE-CONSTRAINED S(v)
V_0	0	
V_1	0	
V_2	1	[IDENTICAL TO ASAP]
V_3	2	
V_4	3	
V_5	4	
V_6	5	
V_7	6	

(e)



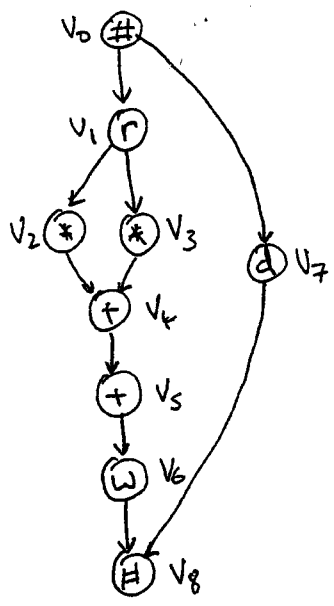
4 (e) [CONTINUED]

NODE v	ASAP $S(v)$	RESOURCE-CONSTRAINED $S(v)$
v_8	0	0
v_0	0	1
v_1	0	2
v_2	0	0
v_3	1	2
v_4	1	1
v_5	2	3
v_6	3	4
v_7	4	5

[4 MARKS]

CYCLE	CANDIDATE SET	SCHEDULED SET	
0	$\{v_0, v_1, v_2\}$	$\{v_2\}$	
1	$\{v_0, v_1, v_4\}$	$\{v_0, v_4\}$	[COULD BE v_1]
2	$\{v_1, v_3\}$	$\{v_1, v_3\}$	
3	$\{v_5\}$	$\{v_5\}$	
4	$\{v_6\}$	$\{v_6\}$	
5	$\{v_7\}$	$\{v_7\}$	

5. a)



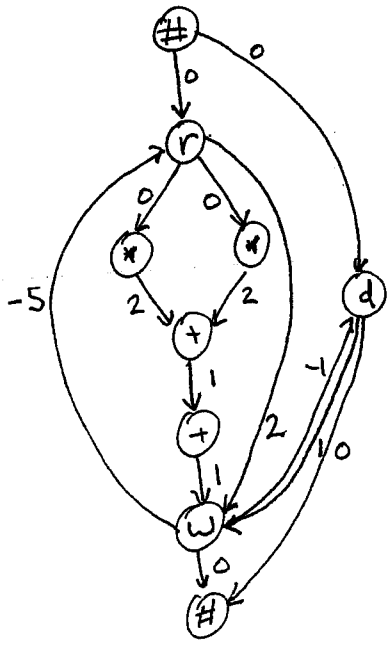
[5 MARKS]

b)

NODE	ASAP
V_0	0
V_1	0
V_2	0
V_3	0
V_4	2
V_5	3
V_6	4
V_7	0
V_8	4

[4 MARKS]

c)



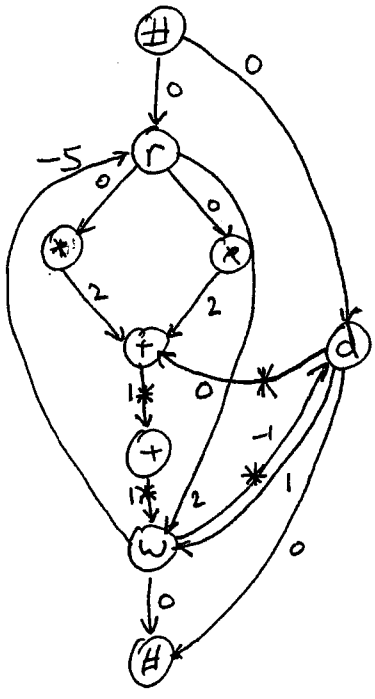
[4 MARKS]

5. (d)

NODE	ASAP
v_0	0
v_1	0
v_2	0
v_3	0
v_4	2
v_5	3
v_6	4
v_7	3
v_8	4

[4 MARKS]

(e)



[1 MARK]

(f) CYCLE MARKED W/ (*) ABOVE HAS POSITIVE TOTAL WEIGHT ⇒ NO LONGEST PATH OF FINITE LENGTH.

[2 MARKS]