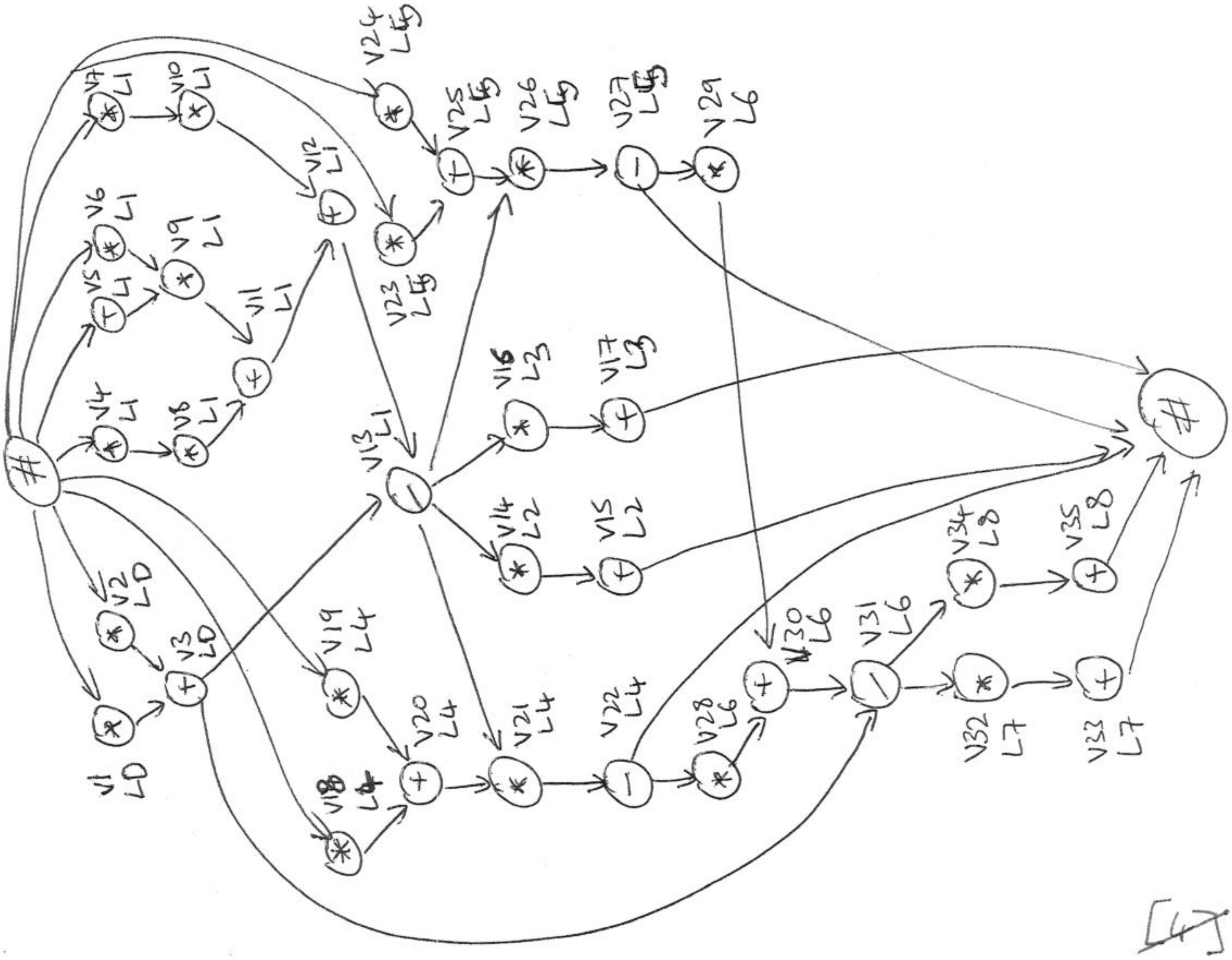
Synthesis of Digital Architectures

Solutions 2008



~~[4]~~

[5]

1.b) Resource constrained list scheduling.

ALAP TIMES (for URGENCY) – coded in reverse, i.e. 0 is high ALAP, greater # ⇒ more urgent.

SOURCE	13	v19	9
v1	10	v20	8
v2	10	v21	7
v3	9	v22	6
v4	12	v23	9
v5	12	v24	9
v6	12	v25	8
v7	11	v26	7
v8	11	v27	6
v9	11	v28	5
v10	10	v29	5
v11	10	v30	4
v12	9	v31	3
v13	8	v32	2
v14	2	v33	1
v15	1	v34	2
v16	2	v35	1
v17	1	SINK	0
v18	9		

SCHEDULE

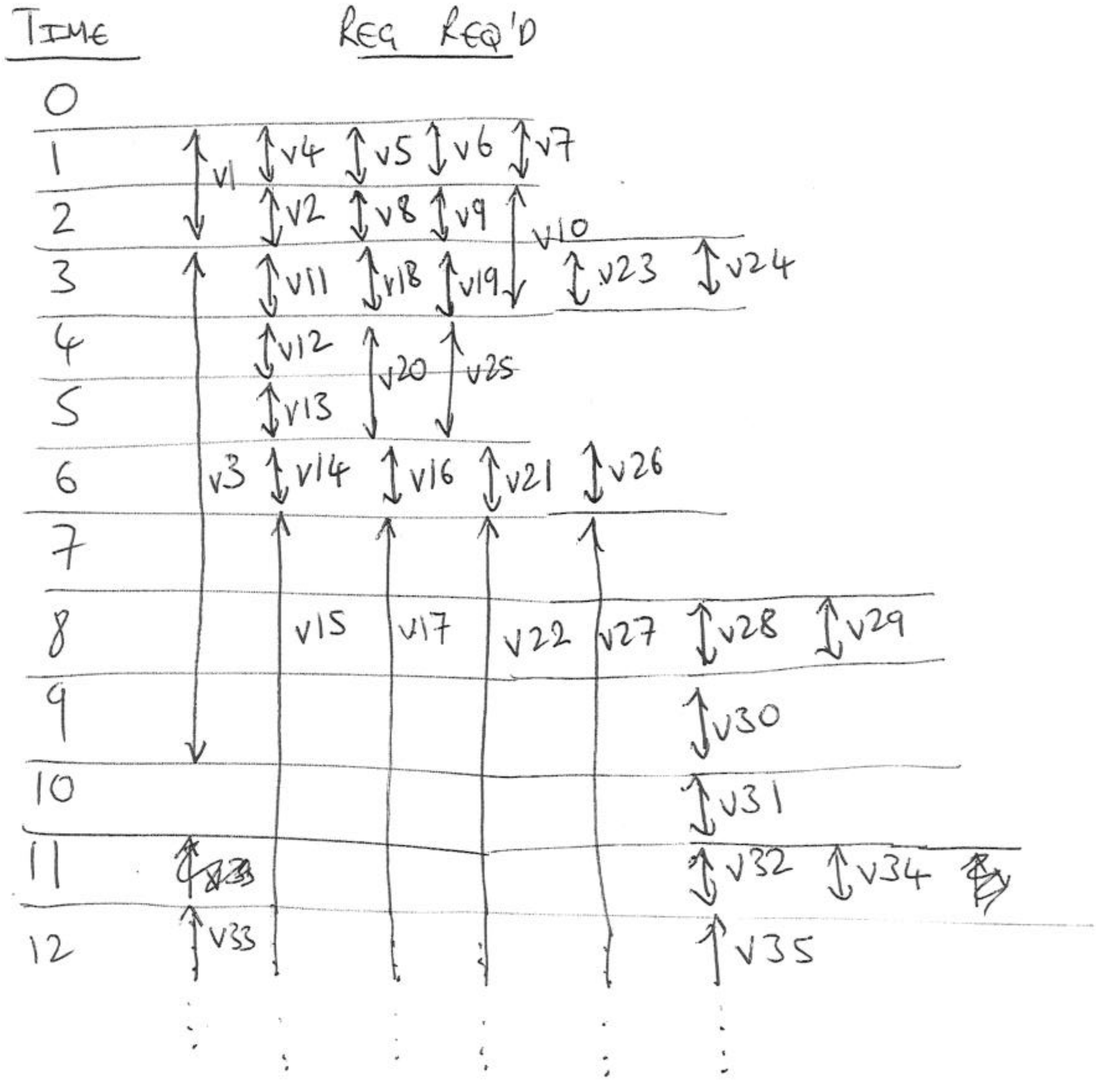
	CANDIDATES	CHOSEN
TIME 0:	v1, v2, v4, v5, v6, v7, v18, v19, v23, v24	v4, v5, v6, v7, v1
1:	v2, v18, v19, v23, v24, v8, v9, v10	v8, v9, v2, v10
2:	v18, v19, v23, v24, v11, v3	v11, v3, v18, v19, v23, v24
3:	v12, v20, v25	v12, v20, v25
4:	v13	v13
5:	v21, v14, v16, v26	v21, v14, v16, v26
6:	v22, v15, v17, v27	v22, v15, v17, v27
7:	v28, v29	v28, v29

1 b) [Continued]

		CAND.	CHOSEN
TIME	8:	v30	v30
	9:	v31	v31
	10:	v32, v34	v32, v34
	11:	v33, v35	v33, v35

~~[4]~~
[5]

c) Consider each node i.d. as the label for its output register. Then



1. c) [Continued]

$\chi(G) = \kappa(G) = 7.$ ~~[4]~~ [5]

d)

Node	Binding	Node	Binding
V_1	$(*,1)$	V_{19}	$(*,2)$
V_2	$(*,4)$	V_{20}	$(+,2)$
V_3	$(+,1)$	V_{21}	$(*,1)$
V_4	$(*,2)$	V_{22}	$(+,1)$
V_5	$(+,1)$	V_{23}	$(*,3)$
V_6	$(*,3)$	V_{24}	$(*,4)$
V_7	$(*,4)$	V_{25}	$(+,3)$
V_8	$(*,1)$	V_{26}	$(*,4)$
V_9	$(*,2)$	V_{27}	$(+,4)$
V_{10}	$(*,3)$	V_{28}	$(*,1)$
V_{11}	$(+,2)$	V_{29}	$(*,2)$
V_{12}	$(+,1)$	V_{30}	$(+,1)$
V_{13}	$(/,1)$	V_{31}	$(/,1)$
V_{14}	$(*,2)$	V_{32}	$(*,1)$
V_{15}	$(+,2)$	V_{33}	$(+,1)$
V_{16}	$(*,3)$	V_{34}	$(*,2)$
V_{17}	$(+,3)$	V_{35}	$(+,2)$
V_{18}	$(*,1)$		

4 $*$

4 $+/-$

1 $\div$

~~[4]~~ [5]

2. a) Min: $\displaystyle\sum_{t=ASAP_v}^{ALAP_v} t \cdot x_{v_z t}$

s.t. $\forall v \in V: \displaystyle\sum_{t=ASAP_v}^{ALAP_v} x_{vt} = 1$

$\forall (u,v) \in E: \displaystyle\sum_{t=ASAP_v}^{ALAP_v} t \cdot x_{vt} \geq \sum_{t=ASAP_u}^{ALAP_u} t \cdot x_{ut} + d_u$

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

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

$$\sum_{r \in R} c_r a_r \leq A$$

[10]

2. b)

$$\text{Min: } \sum_{t=ASAP_{v_z}}^{ALAP_{v_z}} \sum_{q=1}^{4} t \cdot x_{v_z t q}$$

$$\text{s.t. } \forall v \in V \quad \sum_{t=ASAP_v}^{ALAP_v} \sum_{q=1}^{4} x_{vtq} = 1$$

$$\forall (u,v) \in E \quad \sum_{t=ASAP_v}^{ALAP_v} \sum_{q=1}^{4} t \cdot x_{vtq} \geq \sum_{t=ASAP_u}^{ALAP_u} \sum_{q=1}^{4} t \cdot x_{utq} + d_u + \underline{\epsilon_{uv}}$$

communication latency variable

$$\forall v \in V \;\; \forall q \quad \underline{y_{vq}} = \sum_{t=ASAP_v}^{ALAP_v} x_{vtq}$$

placement variable

$$\forall (u,v) \in E \;\; \forall q \quad \epsilon_{uv} \geq y_{vq} - y_{uq}$$ || Ensure $\epsilon_{uv} \geq 1$ if in disjoint quadrants.

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

$$\sum_{v \in V : \tau(v) = r} \; \sum_{t' \in \{t - d_v + 1, \ldots, t\} \cap \{ASAP_v, \ldots, ALAP_v\}} x_{vt'q} \leq \underline{a_{rq}}$$

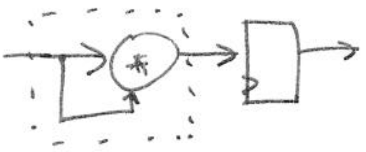
per-quadrant basis.

$$\forall q \quad \sum_{r \in R} c_r a_{qr} \leq A/4$$

[10]

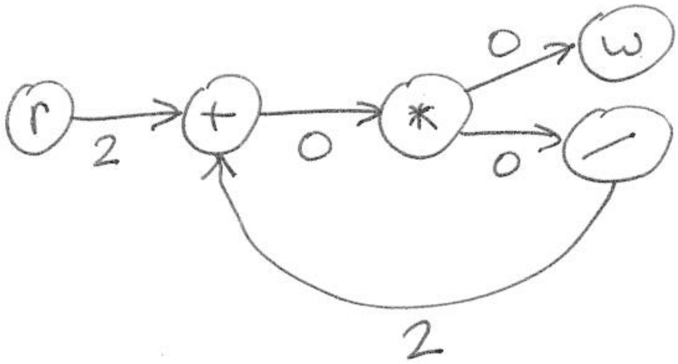
~~8/9~~
7/8

3. a) Need to find an appropriate initial state for registers, e.g.



If this register is initialised to -1, there is no corresponding initial state before the squarer (taking the dotted line as enclosing a "black box". [2]

b)



[3]

c) Min: $L + \alpha \sum_{v \in V} \underline{r_v}$ ← # regs at o/p of node v

s.t. $\cancel{\forall v \in V}\ \cancel{\forall u \in V}\ \forall (v,u) \in E \quad r_v \geq w_r(v,u)$ → NEW CONSTRAINT. CAN RE-USE REGISTERS.

$$\left.\begin{array}{l} \forall (u,v) \in E \quad S_v \geq S_u + d(u) + w_r(u,v) N \\ \forall v \in V \quad S_v + d(v) \leq L \\ \forall (u,v) \in E \quad w_r(u,v) = w(u,v) + r(v) - r(u) \geq 0 \end{array}\right\} \text{As per notes.}$$

$r(v) \in \mathbb{Z}$ for all $v \in V$.

[10]

~~9/9~~
8/8

3 d) Code for (a):

```
while (true)
begin
    read x;
    y = x2 + z1;
    q = y * 3;
    z = 5/q1;
    write q;
    x2 = x1;
    x1 = x;
    z1 = z;
    q1 = q;
end
```

Code for (b):

```
while (true)
begin
    read x;
    ~~y = x1~~
    z = 5/q1;
    y = x1 + z;
    q = y1 * 3;
    write q;
    q1 = q;
    x1 = x;
    y1 = y;
end
```

for (a), N = 4. for (b), N = 3.

For (a), Tclk ≥ 3 (limited by divider)

for (b), Tclk ≥ 4 (divider – adder chain)

~~$\Rightarrow \alpha_2$~~ ~~Both~~ It is clear that $\alpha_2 > \alpha$.

[5]

~~Also $3 + 4\alpha >$~~