Chapter 7 Part 2

Retiming

Instructor: Prof. Peter Lian
Department of Electrical Engineering & Computer Science
Lassonde School of Engineering
York University

CSE4210 Architecture & Hardware for DSP

Solving System of Inequalities

Feasible Retiming Solution

- A solution is feasible if all $w_r(e) \geq 0$, i.e.

  $w_r(e)=w(e)+r(V)-r(U) \geq 0$

  → $r(U)-r(V) \leq w(e)$ for all edges

Example:

$r_1-r_2 \leq 0$

$r_3-r_1 \leq 5$

$r_4-r_1 \leq 4$

$r_4-r_3 \leq -1$

$r_3-r_2 \leq 2$

Solving Systems of Inequalities

- Draw a constraints graph
  - Draw the node i for each of the N variables 1, 2, …, N
  - Draw the node N+1
  - For each inequality $r_i-r_j \leq k$, draw an edge from node j → i with weight k
  - For each node i=1,2,..N draw an edge N+1 → i with weight 0
- Solve
  - The system has a solution if the constraints graph has no negative cycle. Bellman Ford Algorithm
  - One solution is the minimum length from node N+1 to i

Activity 1

- Given the following inequalities, draw the constraint graph.

$r_1 - r_2 \le 0$

$r_3 - r_1 \le 5$

$r_4 - r_1 \le 4$

$r_4 - r_3 \le -1$

$r_3 - r_2 \le 2$

CSE4210 Architecture & Hardware for DSP

Retiming for Clock Period Minimization

Retiming for Clock Period Minimization

- The minimum clock time is the computation time of the critical path.
- Critical path is the path with the longest computation time and **no delay**.
- Retiming could be used to minimize clock period.

Minimize Clock Period

- Minimum feasible clock period of a graph G is $\Phi(G)=\max\{t(p): w(p)=0\}$
- W(U,V) is the minimum number of registers on any path from U → V
- D(U,V) is the maximum computation time among all paths from U → V with weight W(U,V)

Steps in Minimize Clock Period

1. Let $M=t_{max}n$, where t_{max} is the maximum computation time of any node in G, n =number of nodes in G
2. Form a new graph G' which is the same as G except the edge weights are replaced by $w'(e)=Mw(e)-t(U)$ ($e=U \rightarrow V$), where $t(U)$ is computation time of node U.
3. Solve for all-pairs shortest path on G' (S_{UV})
4. If $U \neq V$, then $W(U,V)=\lceil S_{UV}/M \rceil$ and $D(U,V)=M \times W(U,V)-S_{UV}+t(V)$. $\lceil X \rceil$ denotes the smallest integer not less than X.
5. If $U=V$, $W(U,V)=0$, $D(U,V)=t(U)$

Steps in Minimize Clock Period

6. Use $W(U,V)$, $D(U,V)$ to find if there a retiming solution such that $\Phi(G) \leq$ c (cycle time).

   This is done by constructing the following set of constraints

   - **Feasibility constraints**

   $r(U)-r(V) \leq w(e)$ for every edge in G

   - **Critical path constraint**

   $r(U)-r(V) \leq W(U,V)-1$ for all nodes U,V in G such that $D(U,V)>c$.

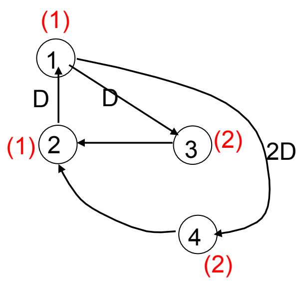
   - If there is a solution to the inequalities (constraints), then the solution is a feasible retiming solution that the circuit can be clocked with period c.

Example

Step 1: find *M*

n=4, t_{max}=2

M=2*4=8

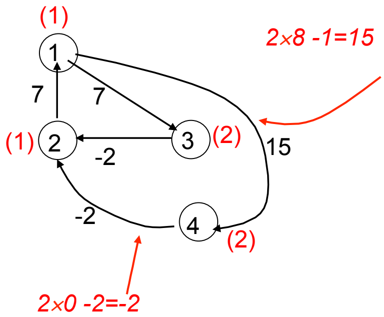


y(n)=ay(n-2) + by(n-3) +x (n)

Step 2: form new graph G'

e = U → V

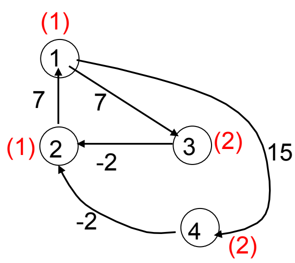
w'(e)=M × w(e)-t(U)



Example

Step 3: Solve all shortest path on G'

S_{UV}	1	2	3	4
1	12	5	7	15
2	7	12	14	22
3	5	-2	12	20
4	5	-2	12	20



Example

Steps 4 and 5: Construct tables for W(U,V) and D(U,V)

S_{UV}	1	2	3	4
1	12	5	7	15
2	7	12	14	22
3	5	-2	12	20
4	5	-2	12	20

U ≠ V, then $W(U,V)= \lceil SUV/M \rceil$

U=V, W(U,V)=0

W(U,V)	1	2	3	4
1	0	1	1	2
2	1	0	2	3
3	1	0	0	3
4	1	0	2	0

$D(U,V)=M\times W(U,V)-S_{UV}+t(V)$

$U=V \quad T(U)$

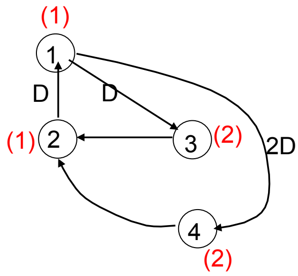
D(U,V)	1	2	3	4
1	1	4	3	3
2	2	1	4	4
3	4	3	2	6
4	4	3	6	2

Example

Step 6: Construct constraints, for c=3

- Feasibility constraints

r(1)-r(3) ≤ 1
r(1)-r(4) ≤ 2
r(2)-r(1) ≤ 1
r(3)-r(2) ≤ 0
r(4)-r(2) ≤ 0



Example

Critical path constraints

$r(U)-r(V) \leq W(U,V)-1$ for all nodes U,V in G

such that $D(U,V)>3$

r(1)-r(2) ≤ 0
r(2)-r(3) ≤ 1
r(2)-r(4) ≤ 2
r(3)-r(1) ≤ 0
r(3)-r(4) ≤ 2
r(4)-r(1) ≤ 0
r(4)-r(3) ≤ 1

D(U,V)	1	2	3	4
1	1	4	3	3
2	2	1	4	4
3	4	3	2	6
4	4	3	6	2

W(U,V)	1	2	3	4
1	0	1	1	2
2	1	0	2	3
3	1	0	0	3
4	1	0	2	0

Example

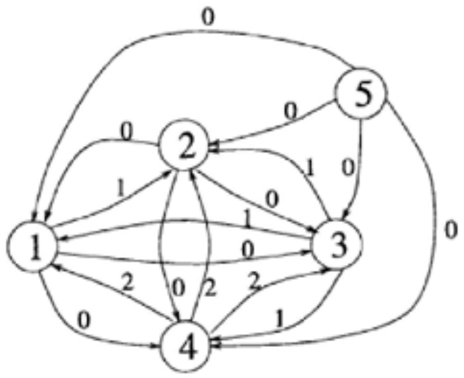
- Combine two sets of constraints, we have 12 inequalities.
  - Note that there is no overlap between these two sets of constraints

Feasibility constraint	Critical path constraint
r(1)-r(3) ≤ 1	r(1)-r(2) ≤ 0
r(1)-r(4) ≤ 2	r(2)-r(3) ≤ 1
r(2)-r(1) ≤ 1	r(2)-r(4) ≤ 2
r(3)-r(2) ≤ 0	r(3)-r(1) ≤ 0
r(4)-r(2) ≤ 0	r(3)-r(4) ≤ 2
	r(4)-r(1) ≤ 0
	r(4)-r(3) ≤ 1

Example

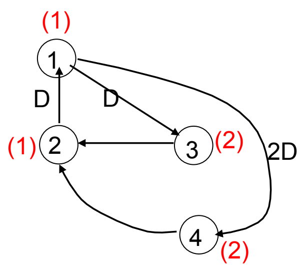
- Solve 12 inequalities
  - Construct constraint graph
  - Find weight matrix W of constraint graph, then solve using Bellman algorithm (Appendix A)

W=[0 1 0 0
0 0 0 0
1 1 0 1
2 2 2 0]



Example

- Solution from Bellman-Ford algorithm:
- r(1)=r(2)=r(3)=r(4)=0, no retiming needed. The graph already has a critical path =3



Redoing for c=2

Critical path constraints

r(U)-r(V) ≤ W(U,V)-1 for all nodes *U,V* in *G* such that *D(U,V)>2*

r(1)-r(2) ≤ 0
r(2)-r(3) ≤ 1
r(2)-r(4) ≤ 2
r(3)-r(1) ≤ 0
r(3)-r(4) ≤ 2
r(4)-r(1) ≤ 0
r(4)-r(3) ≤ 1
r(1)-r(3) ≤ 0
r(1)-r(4) ≤ 1
r(3)-r(2) ≤ -1
r(4)-r(2) ≤ -1

Feasibility constraint

r(1)-r(3) ≤ 1
r(1)-r(4) ≤ 2
r(2)-r(1) ≤ 1
r(3)-r(2) ≤ 0
r(4)-r(2) ≤ 0

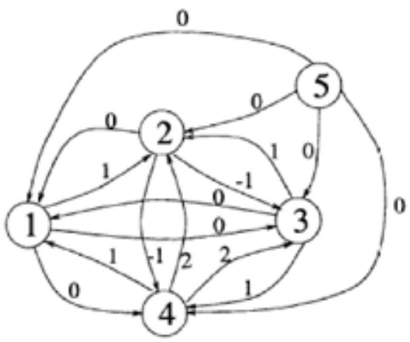
D(U,V)	1	2	3	4
1	1	4	**3**	**3**
2	2	1	4	4
3	4	**3**	2	6
4	4	**3**	6	2

W(U,V)	1	2	3	4
1	0	1	1	2
2	1	0	2	3
3	1	0	0	3
4	1	0	2	0

For C=2

- Solve 12 inequalities
  - Construct constraint graph
  - Find weight matrix W of constraint graph, then solve using Bellman algorithm (Appendix A)

W=[0 1 0 0
0 0 -1 -1
0 0   0 1
1 2   2 0]



For c=2

- Bellman algorithm gives the following solution:

  r(2)=0, r(1)=r(3)=r(4)=-1