A Systematic Approach of Digit-Serial Signal Processing Architecture

Part-II

NTUEE 吳安宇

For

Advanced VLSI Course
Outline

- Introduction
- Simple Examples
- Iteration Bound
- Unfolding
- Systematic Approach
- Properties of Digital-Serial
- Digit-Serial Processing Architecture Designs
- Conclusion
Insert Switch in Unfolding Design

- The following assumptions are made when unfolding an edge $U \rightarrow V$:
  - The wordlength $W$ is a multiple of the unfolding factor $J$, i.e. $W = W'J$.
  - All edges into and out of the switch have no delays.
- With the above two assumptions an edge $U \rightarrow V$ can be unfolded as follows:
  - Write the switching instance as $Wl + u = J( W'l + \lfloor u/J \rfloor ) + (u\%J)$
  - Draw an edge with no delays in the unfolded graph from the node $Uu\%J$ to the node $Vu\%J$, which is switched at time instance $( W'l + \lfloor u/J \rfloor )$. 
Insert Switch in Unfolding Design

Example:

12/ + 1, 7, 9, 11

Unfolding by 3

4/ + 3

U_0 \rightarrow \rightarrow V_0

4/ + 0, 2

U_1 \rightarrow \rightarrow V_1

4/ + 3

U_2 \rightarrow \rightarrow V_2

To unfold the DFG by J=3, the switching instances are as follows:

12/ + 1 = 3(4/ + 0) + 1
12/ + 7 = 3(4/ + 2) + 1
12/ + 9 = 3(4/ + 3) + 0
12/ + 11 = 3(4/ + 3) + 2
Insert Switch (6l+1, 4, 5)

- unfolding by 3
  \[ 6l+1 = 3(2l+0) + 1 \]
  \[ 6l+4 = 3(2l+1) + 1 \]
  \[ 6l+5 = 3(2l+1) + 2 \]

- unfolding by 4!??
  - GCD(6,4)=2, 12l+1, 4, 5, 7, 10, 11
    \[ 12l+1 = 4(3l+0) + 1, 12l+7 = 4(3l+1) + 3 \]
    \[ 12l+4 = 4(3l+1) + 0, 12l+10 = 4(3l+2) + 2 \]
    \[ 12l+5 = 4(3l+1) + 1, 12l+11 = 4(3l+2) + 3 \]
DFG representation of the bit-serial adder with wordlength of 4

Systematic unfolding of the DFG with unfolding factor 2
Property of Unfolding

Unfolding preserves the number of delays in a DFG

\[ U_i \rightarrow V_{(i+w) \% J}, \quad i = 0,1,2 \ldots J - 1 \]

\[
\left\lfloor \frac{w}{J} \right\rfloor + \left\lfloor \frac{w + 1}{J} \right\rfloor + \ldots + \left\lfloor \frac{w + J - 1}{J} \right\rfloor = w
\]
**Property of Unfolding (cont.)**

- J-unfolding of a *loop* $l$ with $w$ delays in the original DFG leads to $gcd(w,J)$ loops in the unfolded DFG.

- Each of these $gcd(w,J)$ loops contains $w/gcd(w,J)$ delays and $J/gcd(w,J)$ copies of each node that appears in $l$. 
Properties of Unfolding (cont.)

- Unfolding a DFG with iteration bond $T$ results in a $J$-unfolded DFG with iteration bond $JT$

\[
T = \max \left\{ \frac{t_l}{w_l} \right\}
\]

\[
T' = \max \left\{ \frac{J / (\gcd(w_l, J)) t_l}{w_l / (\gcd(w_l, J))} \right\} = J \max \left\{ \frac{t_l}{w_l} \right\} = JT
\]
Unfolding Algorithm

2-unfolded DFG

\[ C_1 \rightarrow D_{(9+1)\%2} \text{ with } \left\lfloor \frac{9+1}{2} \right\rfloor \text{ delays} \]

\[ C_0 \rightarrow D_{(9+0)\%2} \text{ with } \left\lfloor \frac{9+0}{2} \right\rfloor \text{ delays} \]
Unfolding Algorithm

2-unfolded

\[ A_0 \rightarrow B_0 \rightarrow A_2 \rightarrow B_2 \rightarrow A_4 \rightarrow B_4 \rightarrow \ldots \]
\[ A_1 \rightarrow B_1 \rightarrow A_3 \rightarrow B_3 \rightarrow A_5 \rightarrow B_5 \rightarrow \ldots \]

2 nodes & 2 edges
\[ T_\infty = (1+1)/2 = 1u^t \]

4 nodes & 4 edges
\[ T_\infty = 2 / 2 = 1u^t \]
Multiplier

<table>
<thead>
<tr>
<th>X7</th>
<th>X6</th>
<th>X5</th>
<th>X4</th>
<th>X3</th>
<th>X2</th>
<th>X1</th>
<th>X0</th>
</tr>
</thead>
<tbody>
<tr>
<td>a3</td>
<td>a2</td>
<td>a1</td>
<td>a0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X7a0</th>
<th>X6a0</th>
<th>X5a0</th>
<th>X4a0</th>
<th>X3a0</th>
<th>X2a0</th>
<th>X1a0</th>
<th>X0a0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X7a1</th>
<th>X6a1</th>
<th>X5a1</th>
<th>X4a1</th>
<th>X3a1</th>
<th>X2a1</th>
<th>X1a1</th>
<th>X0a1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PS1_7</th>
<th>PS1_6</th>
<th>PS1_5</th>
<th>PS1_4</th>
<th>PS1_3</th>
<th>PS1_2</th>
<th>PS1_1</th>
<th>PS1_0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X7a2</th>
<th>X6a2</th>
<th>X5a2</th>
<th>X4a2</th>
<th>X3a2</th>
<th>X2a2</th>
<th>X1a2</th>
<th>X0a2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PS2_7</th>
<th>PS2_6</th>
<th>PS2_5</th>
<th>PS2_4</th>
<th>PS2_3</th>
<th>PS2_2</th>
<th>PS2_1</th>
<th>PS2_0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X7a3</th>
<th>X6a3</th>
<th>X5a3</th>
<th>X4a3</th>
<th>X3a3</th>
<th>X2a3</th>
<th>X1a3</th>
<th>X0a3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a3</th>
</tr>
</thead>
</table>

Multiplication algorithm in tabular form for an 8-bit by 4-bit multiplication
Digit-Serial Processing Architecture Designs

8x4-bit programmable –coefficient serial-parallel multiplier
Digit-Serial Processing Architecture Designs

A 4-unfolding digit-serial multiplier
Decimators

A decimator with decimation ratio of three

A 2-unfolded decimator

A 3-unfolded decimator
Interpolators

A interpolator with interpolation ratio of four

A 3-unfolded interpolator
Advantages of Digit-Serial

- Periodic Schedule Comparison
  - Non-overlapped schedule:
    - the period of the schedule = the critical path
  - Overlapped schedule:
    - the period of the schedule < the critical path
- Constructing overlapped schedules is more complex than the non-overlapped ones!
Advantages of Digit-Serial

(a) Iteration period = 60 units

(b) 2-unfolded of (a) Iteration period = 45 units
Advantages of Digit-Serial

(c) Retimed (a) Iteration period = 40 units

(d) 2-unfolded of (c): Iteration period = 35 units
Advantages of Digit-Serial

(a) A data flow graph
(b) Periodic schedule for iteration period of 8 time units using 4 processors
(c) 2 unfolded version of (b), but only use 3 processors
Advantages of Digit-Serial

❖ Speed Comparison

Bit-serial system:

Sample-rate

\[ f_{bs} = \frac{1}{W(t_d + t_s)} \]

Digit-serial system:

Sample-rate

\[ f_{ds} = \frac{1}{W'T_{ds}} \]

Clock period is 

\[ T_{ds} = t_d + Jt_s \]

\( t_d \) is the sum of the clock rise and fall times and the propagation delay through the latch. 
\( t_s \) is the propagation delay in computation of the sum output of a binary adder.
Conclusion

- Reduce the iteration period
- Reduce the number of clock cycles for an output sample
- Fit in with different design specifications
- High-speed or low power application
References

- Parhi
  - Ch2, Ch5, Ch6

- Paper