NonLinear Pipelines

1. Introduction, Reservation Table, ICV, ... ✓ covered fully
2. Greedy Initiator Controller Design ✓ covered fully
   SKIP state diagrams & MAL analysis
3. Data Stationary Controller Design just mentioned.
   EE560 topic
4. Data path design very little
5. Examples of Datapath & Corresponding
   Reservation Tables ✓ covered fully
Reproduced on the next page

is one of the slides (19/36) from

EE457_NonLinear_P2.jnt

Design Variation #2

Question: If the clock is slow and the combinational cascade does create timing problems, then does it work?
Given on the side are two design variations of the above design.

In the original design, the result of the previous iteration produced by stage $S_3$ (S3/S4) is fed into stage $S_2$ (S1/S2).

In the design variation #1, the result of the previous iteration produced by stage $S_3$ (S3/S4) is fed into stage $S_2$ (S1/S2).

State if the variations are acceptable or not. If unacceptable, state your reasons.

In variation #2, stage 3 result goes into stage 2 as input without passing through any stage register. So it amounts to do combinational cascading of two stages producing a "big FAT stage!"
Answer: True it is confusing. To make it easy to analyze, it is best to redraw the datapath replicating the S3 XOR gates in S2 also as shown on the next page.

The reservation table for this redrawn design is

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>S1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

Conclusion: Yes it works, but the reservation table is quite different.
Design Variation #2

S0: Bit generator/Shift register

S1

S2

S3

S4

Replicated gates of S3