Register Organizations in HLS

Disjoint register organization (mainly non-commutative ops)

Shared register organization (only commutative ops)

Register Organizations for FUs in HLS
Register Allocation in HLS

Use the commutative register organization

Lifetime of variable = [production time, finish time of target operation]

Use graph coloring to obtain minimum # of registers

Variable Lifetime Conflict/Overlap Graph

Schedule for pipelined execution:

Register Allocation in HLS
A second schedule:

Variable Lifetime Conflict/Overlap Graph

Schedule 2 for pipelined execution:

A second schedule
Static Timing Analysis (STA)

Input: DAG $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with node and/or arc delays. Output: Critical path delays

Forward-Trace Process:

1. Let $\ell$ be the # of levels of $\mathcal{G}$ from $0$ to $\ell$ w/ $0 \leftrightarrow \ell$, $\ell \in \mathcal{V}$
2. For every input (level-0) node $n$, is delay of $n$'s inputs at time 0 assumed to be available
3. For every output node $n$:
   a. $\max_{n \in \mathcal{V}} (n)do$ = $(n)do$ [2]
   b. For every output node $n$:
      i. Let $\ell$ be the # of levels of $\mathcal{G}$ from $0$ to $\ell$ w/ $0 \leftrightarrow \ell$
      ii. For every output node $n$, max-path delay through $n$ is the time when the output from $n$ would be available (assuming no hardware constraints), i.e.,
      $\max_{n \in \mathcal{V}} (n)do$

4. For every output $n$:
   a. $\max_{n \in \mathcal{V}} (n)do$ = $(n)do$
   b. For every output node $n$, max-path delay through $n$ is the time when the output from $n$ would be available (assuming no hardware constraints), i.e.,
   $\max_{n \in \mathcal{V}} (n)do$

5. Critical path delay $= \max_{n \in \mathcal{V}} (n)do$

Forward Trace Process:

Input: DAG $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with node and/or arc delays. Output: Critical path delays

1. For every level-$i$ node $n$:
   a. Let $\ell = 0$ to $\ell - 1$
   b. For every level-$i$ node $n$:
      i. Let $\ell$ be the # of levels of $\mathcal{G}$ from $0$ to $\ell$ w/ $0 \leftrightarrow \ell$
      ii. For every output node $n$, max-path delay through $n$ is the time when the output from $n$ would be available (assuming no hardware constraints), i.e.,
      $\max_{n \in \mathcal{V}} (n)do$
the largest output delay gives the actual critical path.

10. Tracking back through the $\text{max-delay}(n)$'s from the output node with

9. End for

8. End for

$(n)^{x_{\text{max}}}P = (z)d_0$ where $z$ is the level $T - 1$ node s.t. $z = \text{max-delay}(n)$

7. If $0 < T$ then

6. $\forall (a^n) \in \mathcal{A}$

5. $\forall (a^n) \in \mathcal{F}$

4. $\forall (a^n) \in \mathcal{W}$

3. For every level-$i$ node $u$ do:

2. For $i = T$ down to 0 do:

End for

1. For each output node $n$ do

Backward Trace Process:

Static Timing Analysis (STA) (Contd)