Unfolding

Version	@March 2, 2022
Property	

Concept

old program ——J-unfolding——> new program（J copies with different execute iterations)

Usage:

- Decrease iteration period (or sample period) to get close to iteration boundary
- Similar to Loop Unrolling

How to Unfold a DFG?

process:

- j copies for nodes from $X_0 ... X_j$, edges from $E_0 ... E_j$
- calculate new weights of $U_i \rightarrow V_{(i+w)\%j}$ by equations: (W_{new} is one of the j copies of an original edge)

$$W_{new} = floor\{(i+w)/j\}$$

Properties

1. **Maybe increase critical path**

   Unfolding will decompose the original edges with weight into j sub-edges. If the $w < J$, the unfolding result will have $j - w$ edges with no weights. Critical path will increase.

Otherwise, if $w \geq J$, critical path won't increase, will stay the same.

2. **Unfolding rules**

   The original loop l (w) will produce $gcd(w, J)$ loops, each loop contains $w/gcd(w, J)$ weights

3. **Increase iteration boundary**

   $T_\infty \rightarrow JT_\infty$

Application

1. Original DFG has nodes which $T_u > T_\infty$

   In the situation, basic retime and pipeline can't resolve the problem, we can UNFOLD. UNFOLD can increase iteration boundary so even though critical path not reduced, we can increase the Parallelism of the system

2. Iteration Boundary is not an INTEGER

   Do $J = w/gcd(T_{loop}, w_{loop})$ retime and pipeline can hardly resolve to achieve equal time as IB and critical path

3. Reduce sample period