SOFTWARE SYNTHESIS

Implementation of Digital Signal Processing

Sabih H. Gerez
University of Twente

SOFTWARE SYNTHESIS

- Generation of executable code from data-flow graphs: *single-processor schedules*
- Used for:
  - Production software
  - Simulation software

Think of the multiple calls to the *run* method in Arx C++ simulations.

- Based on following paper (all examples are taken from it):
  Bhattacharyya, S.S., R. Leupers and P. Marwedel, *Software Synthesis and Code Generation for Signal Processing Systems*, IEEE Transactions on Circuits and Systems---II, Analog and Digital Signal Processing, Vol.47(9), (September 2000).

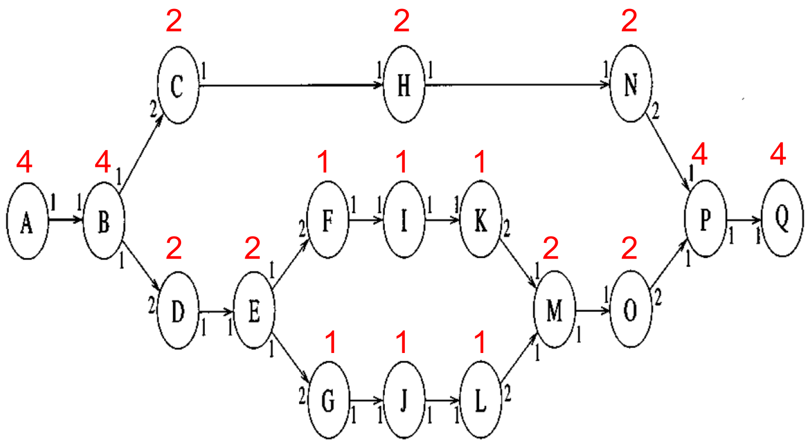
TOPICS

- Synchronous data flow (recap)
- Optimization criteria

SYNCHRONOUS DATA FLOW (SDF)

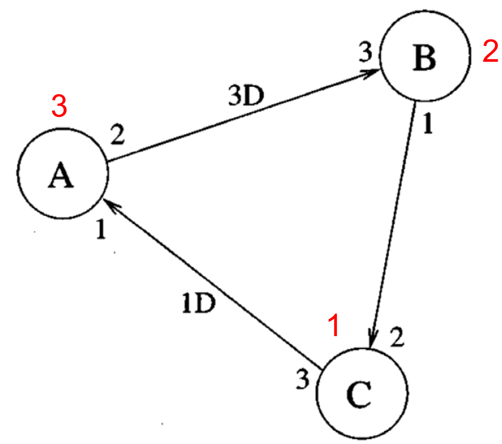
- Already discussed.
- Each *firing* of a node consumes a fixed number of tokens and produces a fixed number of tokens (these numbers are annotated along the edges).
- An edge can have delay (initial tokens).
- Consistency:
  - The *repetitions vector* (relative number of invocations for each node) should exist.
  - There should be no *deadlock* (situation where nodes are waiting for each other to produce tokens).

CONSISTENT SDF EXAMPLE



EXAMPLE OF SDF WITH DEADLOCK

- Easiest check for deadlock: *simulation*



*n*D on an edge means, *n* initial tokens.

4D on edge AB removes deadlock.

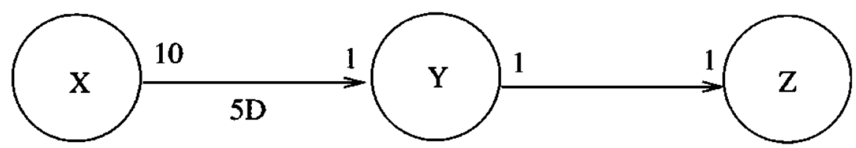
OPTIMIZATION CRITERIA

- Buffer memory
- Code memory
- Number of context switches

IMPLEMENTATION

- Inlined code
- Subroutines
- Hybrid

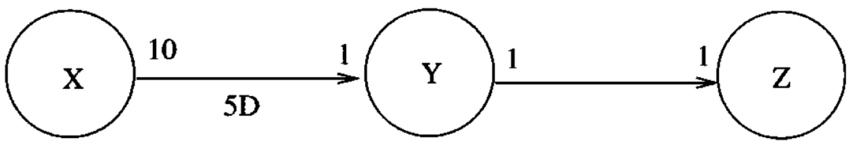
MINIMAL-BUFFER SCHEDULE



$$S_1 = YZYZYZYZYZXYZYZYZYZYZ$$

- Buffer size: $buf(S_1) = 11$
- Code size: $c\_size(S_1) = \kappa(X) + 10\kappa(Y) + 10\kappa(Z)$
- Context switches: $c\_sw(S_1) = 21$

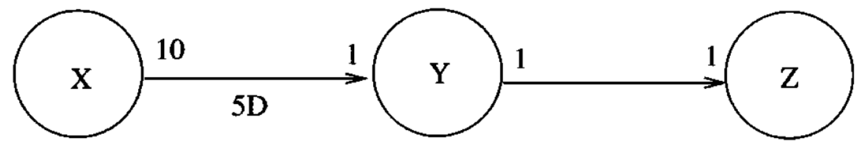
LOOPED SCHEDULE



$$S_2 = (5YZ)X(5YZ)$$

- Buffer size: $buf(S_2) = 11$
- Code size: $c\_size(S_2) \approx \kappa(X) + 2\kappa(Y) + 2\kappa(Z)$
- Context switches: $c\_sw(S_2) = 21$

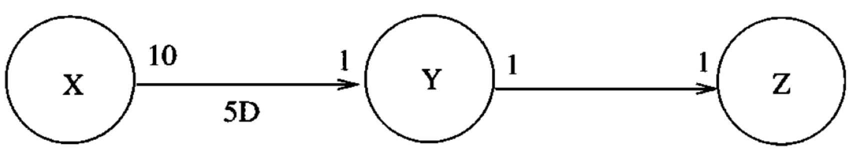
MINIMAL-CODE-SIZE SCHEDULE (1)



$$S_3 = X(10Y)(10Z)$$

- Buffer size: $buf(S_3) = 25$
- Code size: $c\_size(S_3) \approx \kappa(X) + \kappa(Y) + \kappa(Z)$
- Context switches: $c\_sw(S_3) = 3$

MINIMAL-CODE-SIZE SCHEDULE (2)



$$S_4 = X(10YZ)$$

- Buffer size: $buf(S_4) = 16$
- Code size: $c\_size(S_4) \approx \kappa(X) + \kappa(Y) + \kappa(Z)$
- Context switches: $c\_sw(S_4) = 21$