An Introduction to Synchronous Data Flow Model

Dr. Haitao Wei
CAPSL at UDEL

Outline

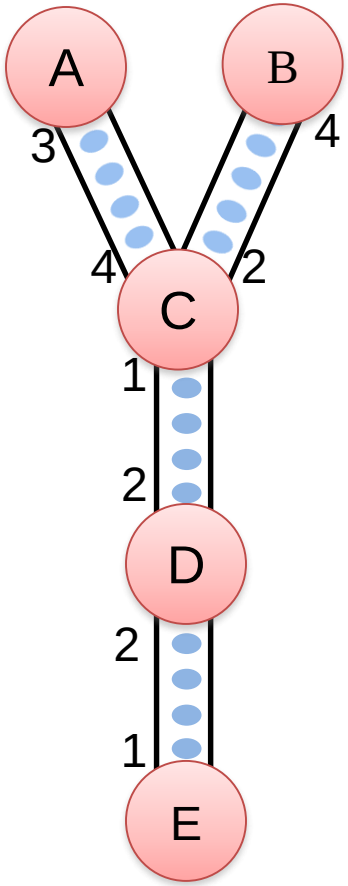
- Synchronous Data Flow Model
  - Definition
  - Example
- Periodic Schedule and Consistency
- Stream Programming Language
  - Structured SDF
- Apply SDF to Bigdata

Outline

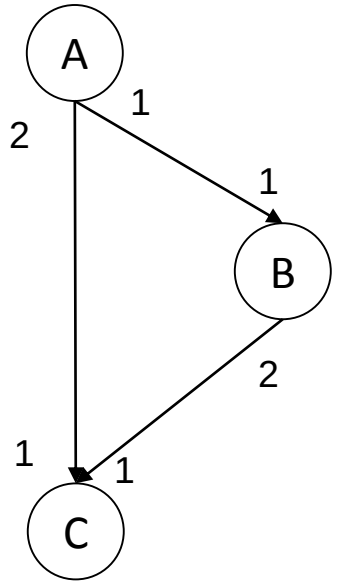
- Synchronous Data Flow Model
  - Definition
  - Example
- Periodic Schedule and Consistency
- Stream Programming Language
  - Structured SDF
- Apply SDF to Bigdata

Synchronous Data Flow Model

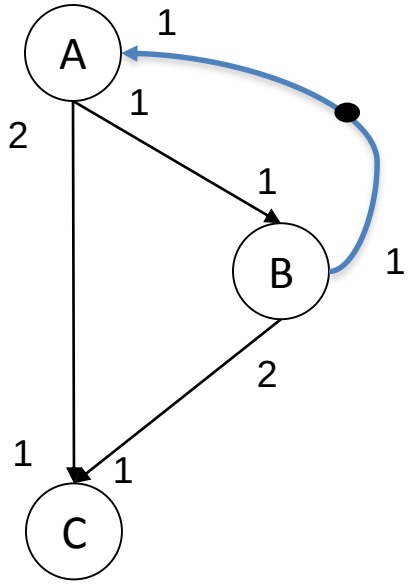
- Synchronous Data Flow (SDF) is represented as a graph
  - Node (actor): Computation
  - Edge: First In First Out (FIFO) Queue
- Each edge has two weights: produce rate and consume rate
- Each edge can also have initial data
- Formal: Tuple<N, E, $E_{p,c,i}$>,
  - N: node
  - E: edge
  - $E_{p,c,i}$: Produce rate, consume rate and initial data



Synchronous Data Flow Model

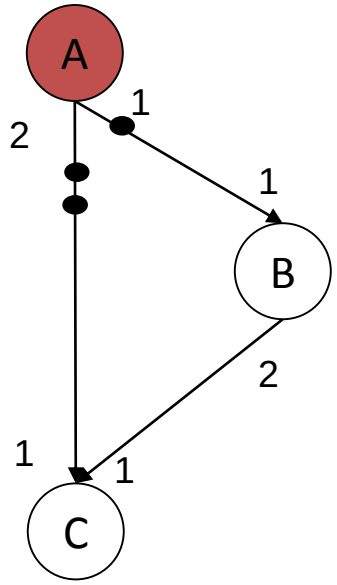


SDF with no initial tokens



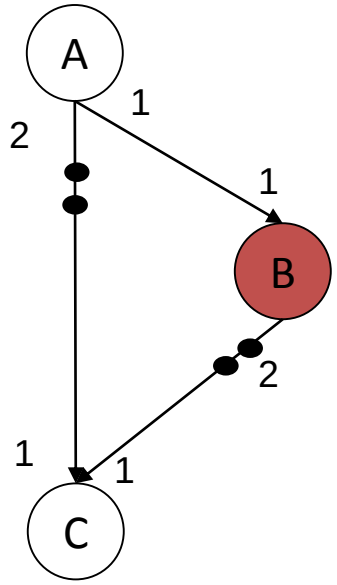
SDF with initial token and loop

Synchronous Data Flow Model



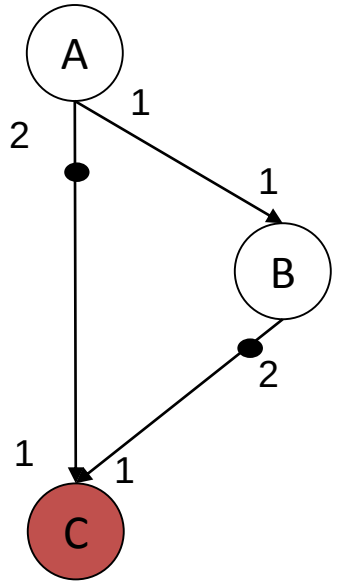
A firing

Synchronous Data Flow Model



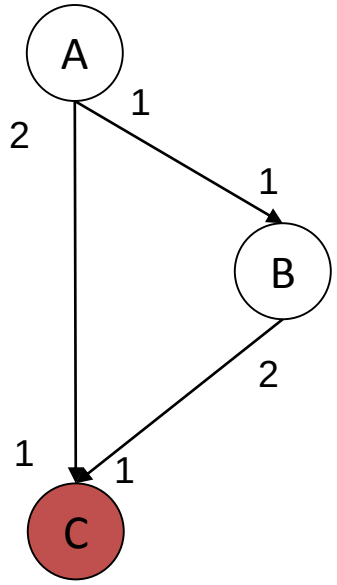
A firing, B firing

Synchronous Data Flow Model



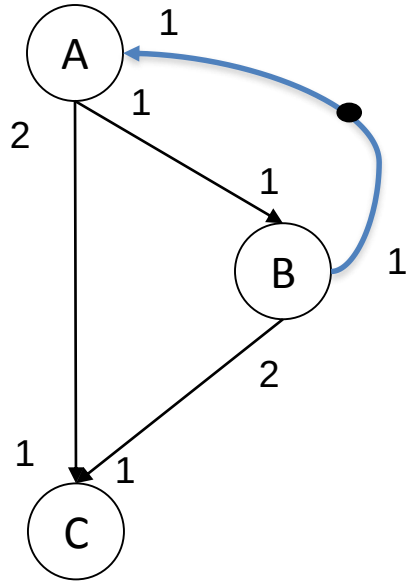
A firing, B firing, C firing

Synchronous Data Flow Model



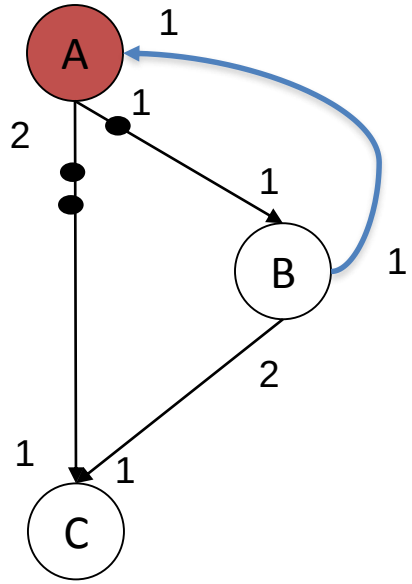
A firing, B firing, C firing, C firing

Synchronous Data Flow Model



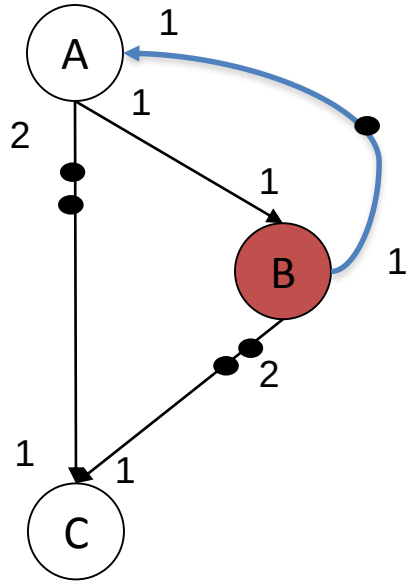
SDF with initial token and loop

Synchronous Data Flow Model



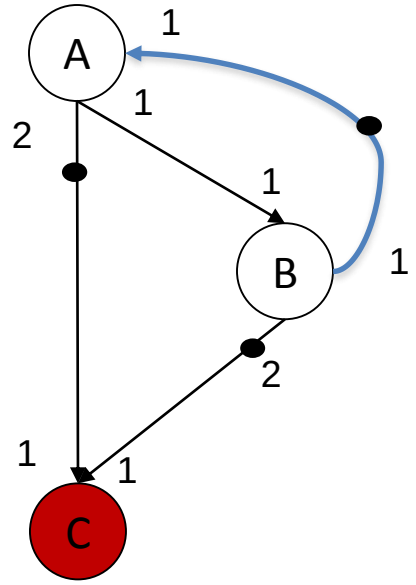
A firing

Synchronous Data Flow Model



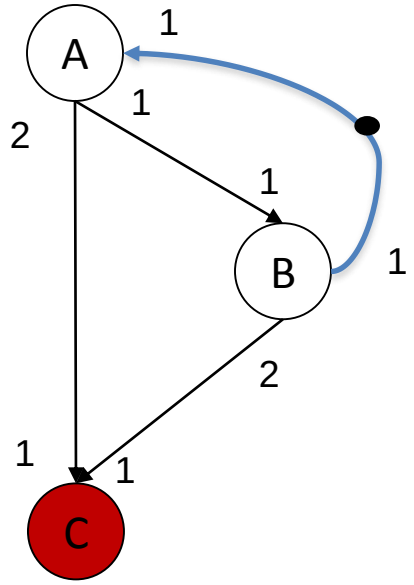
A firing, B firing

Synchronous Data Flow Model



A firing, B firing, C firing

Synchronous Data Flow Model



A firing, B firing, C firing, C firing

Synchronous Data Flow Model

- Question: Can any SDF graph find a firing sequences that makes the state of the graph no changed?
  - State of the graph means: the tokens on each edge are clean, no more no less.
  - Which leads to SDF Consistency Problem

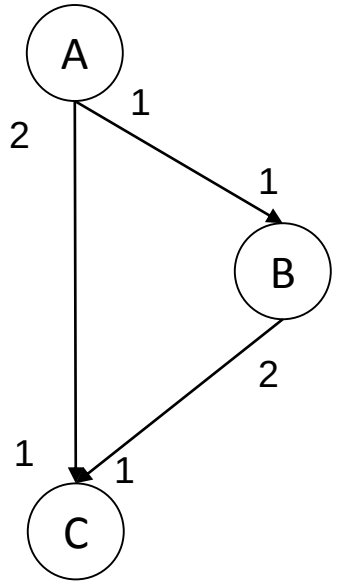
Outline

- Synchronous Data Flow Model
  - Definition
  - Example
- Periodic Schedule and Consistency
- Stream Programming Language
  - Structured SDF in StreamIt
- Apply SDF to Bigdata

Periodic Schedule and Consistency

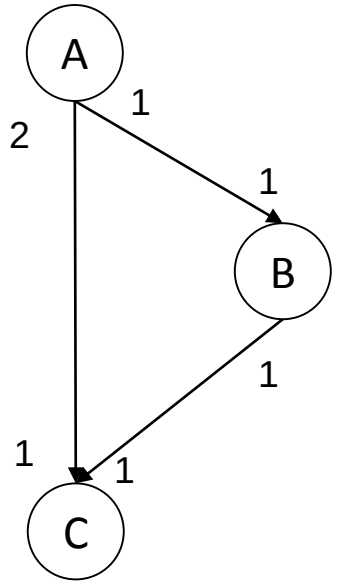
- Firing sequence of a SDF is called a *schedule*
- A *periodic schedule* of an SDF clears all channels and return to its initial status after each node repeats execution a specified finite number of times
- Periodic schedule, permit SDF can process unbounded data with bounded memory
- *A SDF is Consistent* if a periodic schedule exists

Periodic Schedule and Consistency



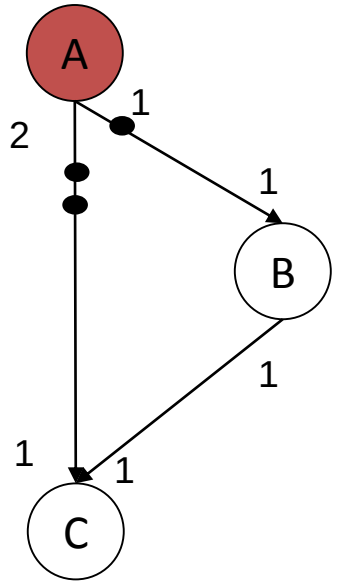
Periodic Schedule: ABCC
AB2C

Periodic Schedule and Consistency



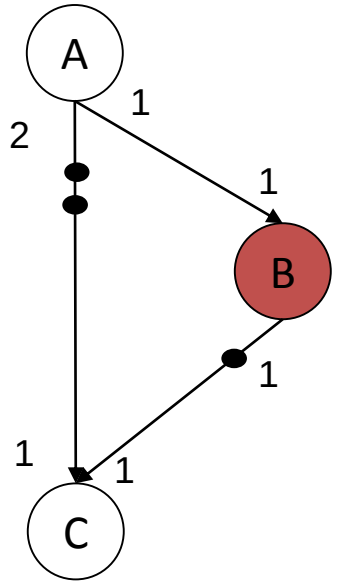
Can you find the periodic schedule?

Periodic Schedule and Consistency



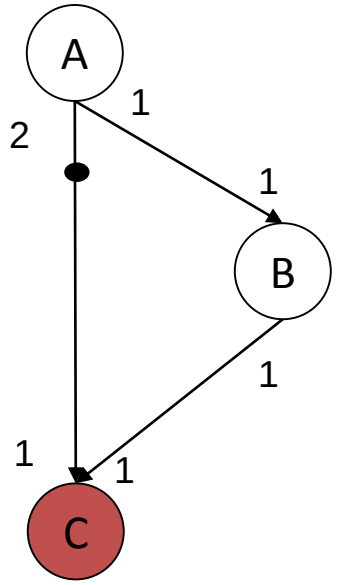
A

Periodic Schedule and Consistency



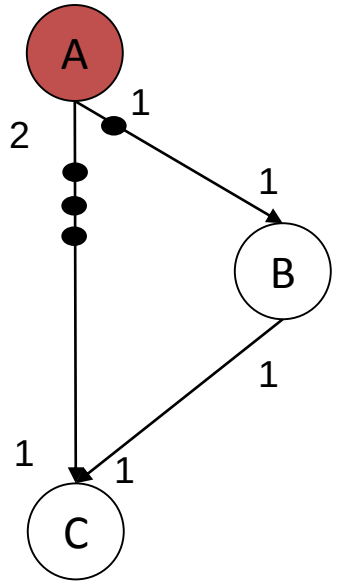
A, B

Periodic Schedule and Consistency



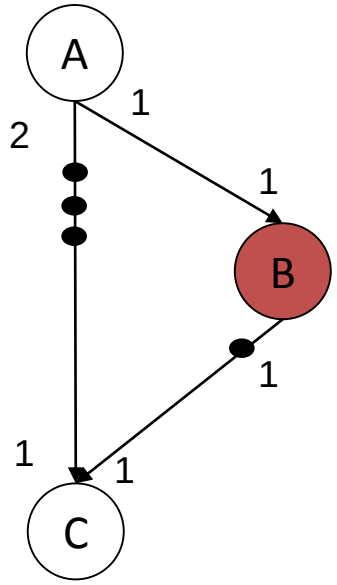
A, B, C

Periodic Schedule and Consistency



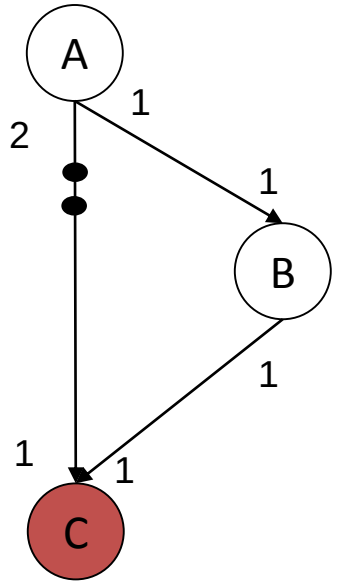
A, B, C, A

Periodic Schedule and Consistency



A, B, C, A, B

Periodic Schedule and Consistency

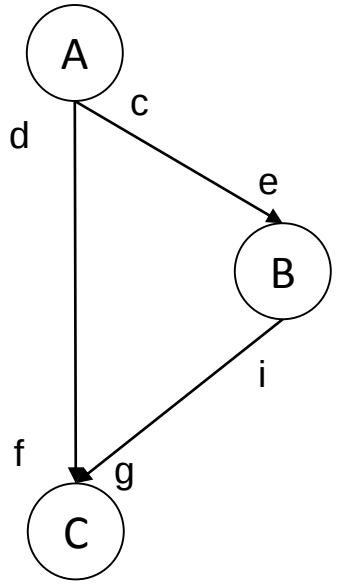


A, B, C, A, B, C

Tokens in channel (A-C) is accumulating which makes the channel unbounded

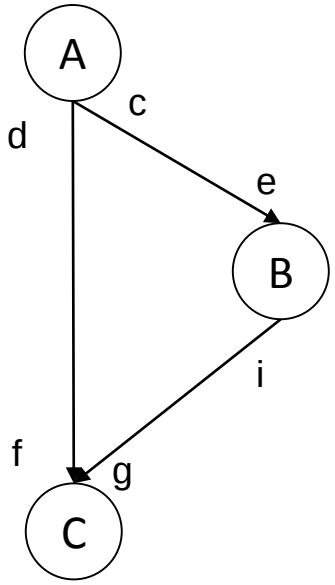
Inconsistent!

Periodic Schedule and Consistency



Problem: Given a general SDF, how can we know it has periodic schedule or not?

Periodic Schedule and Consistency

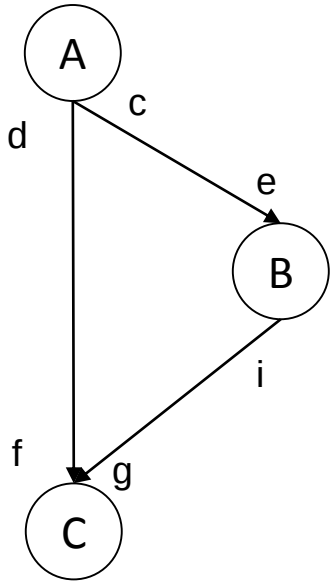


Topology Matrix

- Each row presents the edge
- Each column presents a node
- (i, j): the number of data items placed on i after each invocation of j
- If i is an input channel for j, element (i, j) is negative

$$\begin{array}{ccc} A & B & C \\ \begin{pmatrix} c & -e & 0 \\ d & 0 & -f \\ 0 & i & -g \end{pmatrix} & & \begin{array}{l} A \rightarrow B \\ A \rightarrow C \\ B \rightarrow C \end{array} \end{array}$$

Periodic Schedule and Consistency

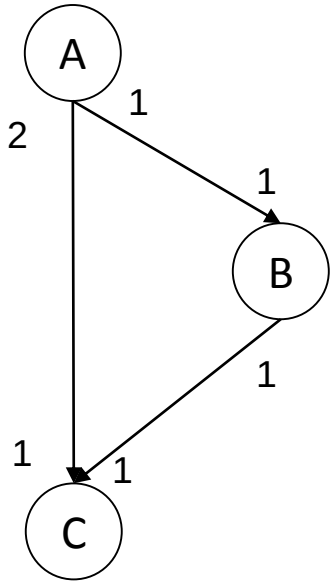


A necessary condition for the existence of a periodic schedule

- the rank of the topology matrix is s − 1, where s is the number of nodes
- Proof: please refer to “Lee’s 87 paper: Synchronous Data Flow”

$$\begin{array}{ccc} A & B & C \\ \begin{pmatrix} c & -e & 0 \\ d & 0 & -f \\ 0 & i & -g \end{pmatrix} & & \begin{array}{l} A\rightarrow B \\ A\rightarrow C \\ B\rightarrow C \end{array} \end{array}$$

Periodic Schedule and Consistency



A necessary condition for the existence of a periodic schedule

- the rank of the topology matrix is s − 1, where s is the number of nodes
- Proof: please refer to “Lee’s 87 paper: Synchronous Data Flow”

A	B	C	
1	−1	0	A→B
2	0	−1	A→C
0	1	−1	B→C

Rank=3 > 2

Outline

- Synchronous Data Flow Model
  - Definition
  - Example
- Periodic Schedule and Consistency
- Stream Programming Language
  - Structured SDF
- Apply SDF to Bigdata

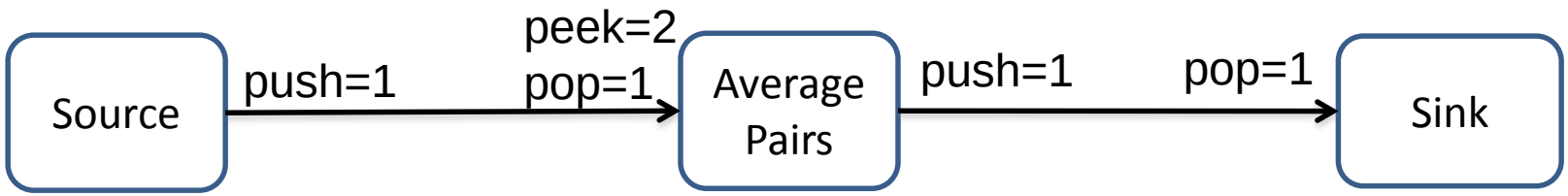
An Stream Example—Average Pairs


```
while ( true ) {
int itm = geneDataItem ();
push(itm);
}
```

```
while(inStream.moreData()) {
  int first=peek(0);
 int second=peek(1);
 push ((first+second)/2);
pop(0);
}
```

```
while (inStream .moreData ())
{
 print(pop(0));
}
```

Average Pairs to Synchronous Dataflow Graph



Extend the SDF to support “peek” sematic

Average Pairs to Synchronous Dataflow Graph

Average Pairs to Synchronous Dataflow Graph

Average Pairs to Synchronous Dataflow Graph

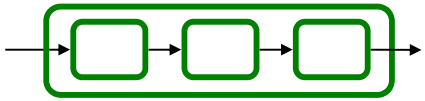
Average Pairs to Synchronous Dataflow Graph

Average Pairs to Synchronous Dataflow Graph

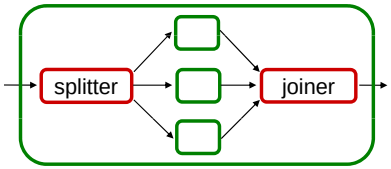
Structured SDF In StreamIt

- Filter

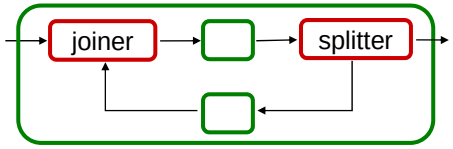
- Pipeline



- Split-Join

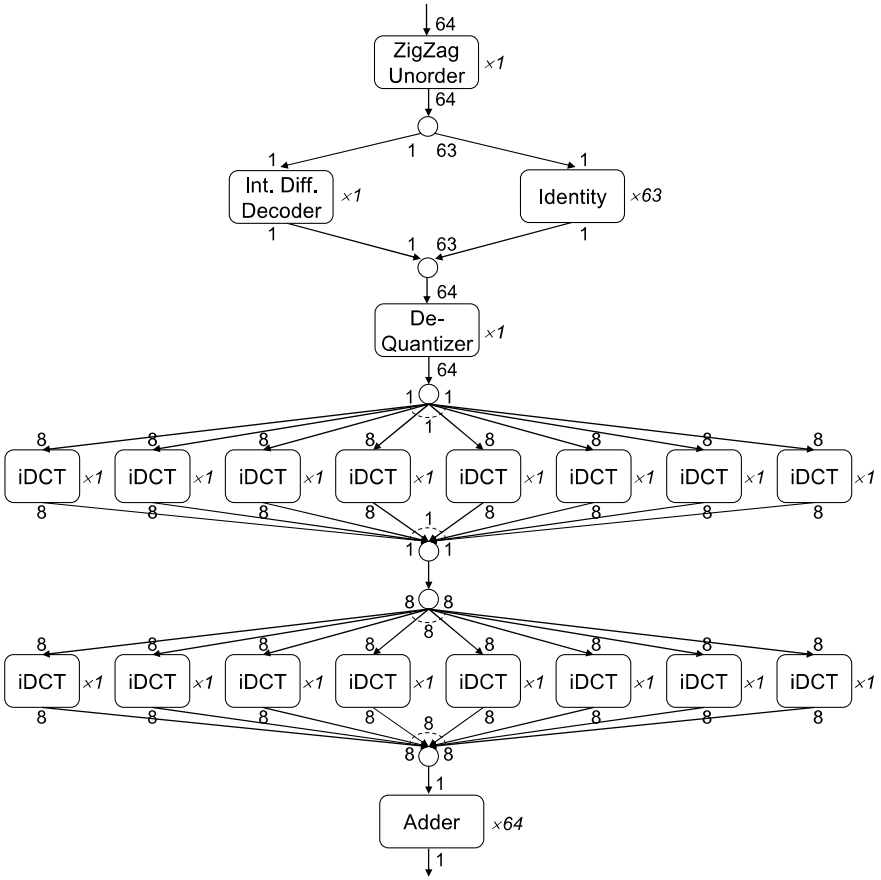


- Feedback Loop



Courtesy by: PACT 2010 paper of streamit: “An Empirical Characterization of Stream Programs and its Implications for Language and Compiler Design ”

Part of JPEG transcoder



Courtesy by: PACT 2010 paper of streamit: “An Empirical Characterization of Stream Programs and its Implications for Language and Compiler Design ”

Weakness of SDF

- Does not support condition (branch)
- Does not support recursion—because it is a static dataflow model
- But still the model is used widely in many application fields

Some Projects Based on SDF model

- Early Ptolemy Project at UC Berkeley
  - Software Synthesis for Embedded system
- StreamIt at MIT
  - streaming program language and compiler
- InforStream and SPL
  - IBM streaming computing product
- Our work COStream
  - hierarchical date flow programming language and compiler
- OpenStream
  - language and compiler support for streaming in OpenMP

Homework

- Write a Fibonacci number generator using Synchronous Data Flow Model
  - Pseudo code for each node in SDF using “peek, push and pop” statements
  - Push Token: PPT animation to show how the tokens flow in SDF graph
  - Periodic Schedule of the SDF

Reference

[1] Early Ptolemy Project at UC Berkeley
- http://ptolemy.eecs.berkeley.edu/projects/index.htm

[2] StreamIt at MIT
- http://groups.csail.mit.edu/cag/streamit/

[3] InforStream and SPL
- http://www-03.ibm.com/software/products/en/infosphere-streams

[4] COStream
- http://www.capsl.udel.edu/pub/doc/papers/dfm12.pdf

[5] OpenStream
- http://openstream.info/