Lecture 7: High-Level Synthesis in the Polyhedral Model

CR11 – Hardware Compilation and Simulation
ENS-Lyon – M2IF

Christophe Alias
High-Level Synthesis

Process Network

- Front-end: dataflow analysis & parallelization
- Back-end: hardware-level scheduling & resource allocation
Polyhedral Process Networks

\[
\text{for } i := 0 \text{ to } N \\
\quad \bullet a[i] = f(i); \\
\text{for } i := 1 \text{ to } N \\
\quad \bullet b[i] := a[i - 1] + a[i];
\]

Partition of the computation: processes

Partition of $\rightarrow_{pc}$: channels $\{\rightarrow_1, \rightarrow_2, \ldots\}$

A schedule $\theta_P$ for each process $P$
for $i := 0$ to $N$
- $a[i] = f(i)$;
for $i := 1$ to $N$
- $b[i] := a[i - 1] + a[i]$;

Partition of the computation: processes
Partition of $\rightarrow_{pc}$: channels $\{\rightarrow_1, \rightarrow_2, \ldots\}$
A schedule $\theta_P$ for each process $P$
for $i := 0$ to $N$
\begin{itemize}
  \item $a[i] = f(i)$;
\end{itemize}
for $i := 1$ to $N$
\begin{itemize}
  \item $b[i] := a[i - 1] + a[i]$;
\end{itemize}

- Partition of the computation: processes
- Partition of $\rightarrow_{pc}$: channels $\{\rightarrow_1, \rightarrow_2, \ldots\}$
- A schedule $\theta_P$ for each process $P$
for $i := 0$ to $N$
  • $a[i] = f(i)$;
for $i := 1$ to $N$
  • $b[i] := a[i - 1] + a[i]$;

Partition of the computation: processes

Partition of $\rightarrow_{pc}$: channels \{ $\rightarrow_1$, $\rightarrow_2$, $\ldots$ \}

A schedule $\theta_P$ for each process $P$
Correctness

\begin{itemize}
  \item \texttt{for } $i := 0 \text{ to } N$
  \item $a[i] = f(i)$;
  \item \texttt{for } $i := 1 \text{ to } N$
  \item $b[i] := a[i - 1] + a[i]$;
\end{itemize}

\textbf{Processes:} any partition

\textbf{Channels:} any partition

\textbf{Schedule:} any schedule increasing along the dependences
for $i := 0$ to $N$

- $a[i] = f(i)$;

for $i := 1$ to $N$

- $b[i] := a[i - 1] + a[i]$;

Give the latency
Latency

\[
\text{for } i := 0 \text{ to } N \\
\bullet \ a[i] = f(i); \\
\text{for } i := 1 \text{ to } N \\
\bullet \ b[i] := a[i - 1] + a[i];
\]

\[
\text{Execution order: } \prec_{PPN} = \rightarrow_{pc} \cup \left( \bigcup_i \prec_{\theta_{P_i}} \right) \\
\text{Latency: } \lambda_{PPN} = 1 + \text{height}(\prec_{PPN})
\]
Hardware Compilation Flow

for \( i := 0 \) to \( N \)

• \( a[i] = f(i) \);

for \( i := 1 \) to \( N \)

• \( b[i] := a[i - 1] + a[i] \);

1. Find the partitions and the schedule
   → minimize the latency

2. Set multiplexors/demultiplexors

3. Select/allocate the channels
   → FIFO preferred

PPN → circuit

1. Factorize the control and the channels

2. Synthesize the processes and the channels
   → with a mainstream HLS tool
Multiplexing

\[
\text{for } i := 0 \text{ to } N \\
\bullet \quad a[i] = f(i); \\
\text{for } i := 1 \text{ to } N \\
\bullet \quad b[i] := a[i - 1] + a[i];
\]

\[
\begin{align*}
\rightarrow & \quad P_1 \quad \rightarrow \\
& \quad \quad \quad \rightarrow \\
& \quad \quad \quad \rightarrow \\
& \quad \quad \quad \rightarrow \\
0 & \quad 1 & \quad 2 & \quad 3 & \quad 4 & \quad 5 \\
& \downarrow & \downarrow & \downarrow & \downarrow & \downarrow
\end{align*}
\]

Definition 1
\[
x \xrightarrow{pc} (y, k) \iff x \xrightarrow{pc} y \text{ on the } k\text{-th read of } y
\]

Theorem 2
\[
\sigma = \xrightarrow{-1}_{pc} \text{ is a function, called the source function.}
\]
Give $\sigma(\langle P_4, i \rangle, 1)$
for $i := 0$ to $N$
- $a[i] = f(i)$;

for $i := 1$ to $N$
- $b[i] := a[i - 1] + a[i]$;

How to implement the channels?
A channel might be implemented by a FIFO iff

- the values are read in the production order (\textit{in-order})
- each value is read exactly once (\textit{unicity})
ILP formulation

In-order communication

\[ \text{in-order}(\rightarrow_c, \prec_P, \prec_C) := \forall x \rightarrow_c x', \forall y \rightarrow_c y' : x' \prec_C y' \Rightarrow x \preceq_P y \]

Read unicity

\[ \text{unicity}(\rightarrow_c) := \forall x \rightarrow_c x', \forall y \rightarrow_c y' : x' \neq y' \Rightarrow x \neq y \]
Quizz

for $i := 0$ to $N$

$LD_A$: load($a[i]$);
$LD_B$: load($b[i]$);

for $i := 0$ to $2N$

$S$: $c[i] := 0$;

for $i := 0$ to $N$
  for $j := 0$ to $N$

$T$: $c[i+j] := c[i+j] + a[i] \cdot b[j]$;

$\theta_T(i,j) = (i,j)$

---

Draw the PPN and specify the channels
On the board!