Type-Directed Scheduling Of Streaming Accelerators – Technical Appendix

David Durst[1], Matthew Feldman[1], Dillon Huff[1], David Akeley[2], Ross Daly[1], Gilbert Bernstein[3], Marco Patrignani[1,4], Kayvon Fatahalian[1], Pat Hanrahan[1]
Stanford University, USA[1] University of California, Los Angeles, USA[2] University of California, Berkeley, USA[3]
CISPA Helmholtz Center for Information Security, Germany[4]

CCS Concepts: • **Hardware** → **Hardware description languages and compilation**; • **Software and its engineering** → **Data types and structures**; *Data flow languages*; • **Computer systems organization** → *Data flow architectures.*

ACM Reference Format:
David Durst[1], Matthew Feldman[1], Dillon Huff[1], David Akeley[2], Ross Daly[1], Gilbert Bernstein[3], Marco Patrignani[1,4], Kayvon Fatahalian[1], Pat Hanrahan[1]. 2020. Type-Directed Scheduling Of Streaming Accelerators – Technical Appendix. In *Proceedings of the 41st ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI '20), June 15–20, 2020, London, UK*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3385412.3385983

This appendix contains additional material from the original paper. Appendix A contains formulas for calculating area and delay for L^{st} operators. Appendix B formalises L^{seq} and Appendix C formalises L^{st}.

A Formulas for L^{st} Operator Properties

For simplicity, we provide only the formulas for the operators used in the code example in the paper.

A.1 Areas of L^{st} operators (excerpts)

Area a is measure of FPGA resources required to implement an operator. Area is a vector of two components, storage and compute, to account for the fact that FPGAs have different resources for storing data and performing computation [2].

`counter_size(n)` computes the area for a counter that counts up to `n`.

$$\texttt{area}\,(\texttt{tuple}) = \{\texttt{compute} = 0, \texttt{storage} = 0\}$$

$$\texttt{area}\,(\texttt{map\_s f}) = n * \texttt{area}\,(\texttt{f})$$

where the input has type `SSeq n t`

$$\texttt{area}\,(\texttt{map\_t f}) = \texttt{area}\,(\texttt{f})$$

$$\texttt{area}\,(\texttt{map2\_s f}) = n * \texttt{area}\,(\texttt{f})$$

where the input has type `SSeq n t`

$$\texttt{area}\,(\texttt{map2\_t f}) = \texttt{area}\,(\texttt{f})$$

$$\texttt{area}\,(\texttt{reduce\_s f}) = (n-1) * area(f)$$

where the input has type `SSeq n t`

$$\begin{aligned}\texttt{area}\,(\texttt{reduce\_t f}) = \;&\texttt{area}\,(\texttt{f}) + \\ &\{\texttt{compute} = 0, \\ &\texttt{storage} = \texttt{sizeof}\,(t)\} + \\ &\texttt{counter\_size(n)}\end{aligned}$$

where the input has type `TSeq n i t`

$$\texttt{area}\,(\texttt{shift\_s}) = \{\texttt{compute} = 0, \texttt{storage} = 0\}$$

$$\texttt{area}\,(\texttt{shift\_t}) = \{\texttt{compute} = 0, \texttt{storage} = \texttt{sizeof}\,(t)\}$$

where the input has type `TSeq n i t`

$$\texttt{area}\,(\texttt{select\_1d\_s}) = \{\texttt{compute} = 0, \texttt{storage} = 0\}$$

$$\texttt{area}\,(\texttt{select\_1d\_t}) = \{\texttt{compute} = 0, \texttt{storage} = 0\}$$

$$\texttt{area}\,(\texttt{reshape}) = \text{see } [1]$$

$$\texttt{area}\,(\texttt{g . f}) = \texttt{area}\,(\texttt{g}) + \texttt{area}\,(\texttt{f})$$

A.2 Delays of L^{st} operators (excerpts).

Delay is a measure of time (in clocks) between the first element of an input sequence arriving at a operator, and the first element emitted by the operator. A fully combinational adder has zero delay. Both the full parallel (`map_s`) and fully sequential `map_t` operators begin emitting output as soon as their contained function `f` does, so the delay of these higher order operators is the same as the delay of `f`.

PLDI '20, June 15–20, 2020, London, UK

ACM ISBN 978-1-4503-7613-6/20/06…$15.00
https://doi.org/10.1145/3385412.3385983

$$
\begin{aligned}
\texttt{delay}\,(\texttt{add}) &= 0\\
\texttt{delay}\,(\texttt{tuple}) &= 0\\
\texttt{delay}\,(\texttt{map\_s}\ \ \texttt{f}) &= \texttt{delay}\,(\texttt{f})\\
\texttt{delay}\,(\texttt{map\_t}\ \ \texttt{f}) &= \texttt{delay}\,(\texttt{f})\\
\texttt{delay}\,(\texttt{reduce\_s}\ \ \texttt{f}) &= 0\\
\texttt{delay}\,(\texttt{reduce\_t}\ \ \texttt{f}) &= n-1\\
&\text{where the input has type } \texttt{TSeq n i t}\\
\texttt{delay}\,(\texttt{shift\_s}) &= 0\\
\texttt{delay}\,(\texttt{shift\_t}) &= 0\\
\texttt{delay}\,(\texttt{select\_1d\_s}) &= 0\\
\texttt{delay}\,(\texttt{select\_1d\_t}) &= j-1
\end{aligned}
$$

where the selecting the jth element

$$
\begin{aligned}
\texttt{delay}\,(\texttt{reshape}) &= \text{see [1]}\\
\texttt{delay}\,(\texttt{f}\ \ .\ \ \texttt{g}) &= \texttt{delay}\,(\texttt{f}) + \texttt{delay}\,(\texttt{g})
\end{aligned}
$$

B $\mathsf{L}^{\mathsf{seq}}$ Formalisation

B.1 Terms

$$
\begin{aligned}
t ::=\ & \texttt{undef} \mid n \in \mathcal{N} \mid b \in \{\texttt{True}, \texttt{False}\}\\
\mid\ & \lambda x : \tau.\ t \mid x \mid [t, \ldots, t] \mid \langle t, \ldots, t\rangle \mid t.i\\
\mid\ & \texttt{tuple\_to\_seq}\ \ t \mid \texttt{seq\_to\_tuple}\ \ t\\
\mid\ & \texttt{not}\ \ t \mid t\ \ \texttt{==}\ \ t \mid t\ \ \texttt{op}\ \ t \text{ s.t. op} \in \{+, -, *, /\}\\
\mid\ & t\ \ \texttt{bop}\ \ t \text{ s.t. bop} \in \{\vee, \wedge\}\\
\mid\ & \texttt{const\_gen}\ t\\
\mid\ & \texttt{shift}\ \ t\ t \mid \texttt{up\_1d}\ \ t\ t \mid \texttt{select\_1d}\ \ t\ t\\
\mid\ & \texttt{partition}\ \ t\ t\ t \mid \texttt{unpartition}\ \ t\\
\mid\ & \texttt{map}\ \ t\ t \mid \texttt{map2}\ \ t\ t\ t \mid \texttt{reduce}\ \ t\ t
\end{aligned}
$$

B.2 Values

$$
v ::= \mathrm{undef} \mid \mathrm{n} \mid \mathrm{b} \mid \lambda x : \tau.\ t \mid [v_1, \ldots, v_n] \mid \langle v, \cdots, v\rangle
$$

B.3 Types

$$\tau ::= \mathbb{N} \mid \mathbb{B}$$

$$\sigma ::= \mathrm{seq}\ n\ \sigma \mid \sigma \times \sigma \mid \tau$$

$$f ::= \sigma \to \sigma \mid \sigma$$

B.4 $\mathsf{L}^{\mathsf{seq}}$ Evaluation Contexts

$$
\begin{aligned}
E ::=\ & [\cdot] \mid E\ t \mid (\lambda x : \tau.\ t)\ E\\
\mid\ & [E, \ldots, t_n] \mid [v_1, \ldots, E, \ldots, t_n] \mid [v_1, \ldots, v_{n-1}, E] \mid \langle E, \ldots, t_n\rangle\\
\mid\ & \langle v_1, \ldots, E, \ldots, t_n\rangle \mid \langle v_1, \ldots, v_{n-1}, E\rangle\\
\mid\ & \texttt{tuple\_to\_seq}\ \ E \mid \texttt{seq\_to\_tuple}\ \ E\\
\mid\ & \texttt{not}\ \ E \mid E \texttt{ == } t \mid v \texttt{ == } E \mid E\ \ \texttt{op}\ \ t \mid n\ \ \texttt{op}\ \ E\\
\mid\ & E\ \ \texttt{bop}\ \ t \mid b\ \ \texttt{bop}\ \ E\\
\mid\ & \texttt{lut\_gen}\ E\ t \mid \texttt{lut\_gen}\ v\ E \mid \texttt{const\_gen}\ E\\
\mid\ & \texttt{shift}\ \ n\ E \mid \texttt{up\_1d}\ \ n\ E \mid \texttt{select\_1d}\ \ n\ E\\
\mid\ & \texttt{partition}\ \ n\ n\ E \mid \texttt{unpartition}\ \ E\\
\mid\ & \texttt{map}\ \ E\ t \mid \texttt{map}\ \ v\ E \mid \texttt{map2}\ \ E\ t\ t \mid \texttt{map2}\ \ v\ E\ t \mid \texttt{map2}\ \ v\ v\ E\\
\mid\ & \texttt{reduce}\ \ E\ t \mid \texttt{reduce}\ \ v\ E
\end{aligned}
$$

B.5 $\mathsf{L}^{\mathsf{seq}}$ Program Contexts

$$
\begin{aligned}
\mathfrak{c} ::=\ & [\cdot] \mid \mathfrak{c}\ t \mid t\ \mathfrak{c} \mid \lambda x : \tau.\ \mathfrak{c}\\
\mid\ & [\mathfrak{c}, \ldots, t_n] \mid [t_1, \ldots, \mathfrak{c}, \ldots, t_n] \mid [t_1, \ldots, t_{n-1}, \mathfrak{c}] \mid \langle \mathfrak{c}, \ldots, t_n\rangle\\
\mid\ & \langle t_1, \ldots, \mathfrak{c}, \ldots, t_n\rangle \mid \langle t_1, \ldots, t_{n-1}, \mathfrak{c}\rangle\\
\mid\ & \texttt{tuple\_to\_seq}\ \ \mathfrak{c} \mid \texttt{seq\_to\_tuple}\ \ \mathfrak{c}\\
\mid\ & \texttt{not}\ \ \mathfrak{c} \mid \mathfrak{c} \texttt{ == } t \mid t \texttt{ == } \mathfrak{c} \mid \mathfrak{c}\ \ \texttt{op}\ \ t \mid n\ \ \texttt{op}\ \ \mathfrak{c}\\
\mid\ & \mathfrak{c}\ \ \texttt{bop}\ \ t \mid t\ \ \texttt{bop}\ \ \mathfrak{c}\\
\mid\ & \texttt{lut\_gen}\ \mathfrak{c}\ t \mid \texttt{lut\_gen}\ t\ \mathfrak{c} \mid \texttt{const\_gen}\ \mathfrak{c}\\
\mid\ & \texttt{shift}\ \ n\ \mathfrak{c} \mid \texttt{up\_1d}\ \ n\ \mathfrak{c} \mid \texttt{select\_1d}\ \ n\ \mathfrak{c}\\
\mid\ & \texttt{partition}\ \ n\ n\ \mathfrak{c} \mid \texttt{unpartition}\ \ \mathfrak{c}\\
\mid\ & \texttt{map}\ \ \mathfrak{c}\ t \mid \texttt{map}\ \ t\ \mathfrak{c} \mid \texttt{map2}\ \ \mathfrak{c}\ t\ t \mid \texttt{map2}\ \ t\ \mathfrak{c}\ t \mid \texttt{map2}\ \ t\ t\ \mathfrak{c} \mid\\
\mid\ & \texttt{reduce}\ \ \mathfrak{c}\ t \mid \texttt{reduce}\ \ t\ \mathfrak{c}
\end{aligned}
$$

C L^{st} Formalisation

C.1 Terms

$$
\begin{aligned}
t ::=\ & \texttt{undef} \mid n \in \mathcal{N} \mid b \in \{\texttt{True}, \texttt{False}\}\\
\mid\ & \lambda x : \tau.\ t \mid x \mid [t, \ldots, t] \mid \langle t, \ldots, t\rangle \mid t.i\\
\mid\ & \texttt{not}\ \ t \mid t\ \ \texttt{==}\ \ t \mid t\ \ \texttt{op}\ \ t \text{ s.t. op} \in \{+, -, *, /\}\\
\mid\ & t\ \ \texttt{bop}\ \ t \text{ s.t. bop} \in \{\vee, \wedge\}\\
\mid\ & \texttt{const\_gen}\ t\\
\mid\ & \texttt{shift\_s}\ \ t\ t \mid \texttt{shift\_t}\ \ t\ t \mid \texttt{up\_1d\_s}\ \ t\ t \mid \texttt{up\_1d\_t}\ \ t\ t\\
\mid\ & \texttt{select\_1d\_s}\ \ t\ t \mid \texttt{select\_1d\_t}\ \ t\ t\\
\mid\ & \texttt{map\_s}\ \ t\ t \mid \texttt{map\_t}\ \ t\ t \mid \texttt{map2\_s}\ \ t\ t\ t \mid \texttt{map2\_t}\ \ t\ t\ t\\
\mid\ & \texttt{reduce\_s}\ \ t\ t \mid \texttt{reduce\_t}\ \ t\ t\\
\mid\ & \texttt{reshape}\ \ \sigma\ \sigma\ t
\end{aligned}
$$

C.2 Values

$$
\begin{aligned}
v ::=\ & \mathrm{undef} \mid \mathrm{n} \mid \mathrm{b} \mid \lambda x : \tau.\ t \mid [v_1, \ldots, v_n]_s \mid \mid [v_1, \ldots, v_n]_t \mid\\
& \langle v, \cdots, v\rangle \mid \mathrm{invalid}
\end{aligned}
$$

C.3 Types

$$\tau ::= \mathbb{N} \mid \mathbb{B}$$

$$\sigma ::= \mathrm{sseq}\ n\ \sigma \mid \mathrm{tseq}\ n\ n\ \sigma \mid \sigma \times \sigma \mid \tau$$

$$f ::= \sigma \to \sigma$$

C.4 L^{st} Evaluation Contexts

$$
\begin{aligned}
E ::=\ & [\cdot] \mid E\ t \mid (\lambda x : \tau.\ t)\ E\\
\mid\ & [E, \ldots, t_n]_s \mid [v_1, \ldots, E, \ldots, t_n]_s \mid [v_1, \ldots, v_{n-1}, E]_s \mid [E, \ldots, t_n]_t\\
\mid\ & [v_1, \ldots, E, \ldots, t_n]_t \mid [v_1, \ldots, v_{n-1}, E]_t\\
\mid\ & \langle E, \ldots, t_n\rangle \mid \langle v_1, \ldots, E, \ldots, t_n\rangle \mid \langle v_1, \ldots, v_{n-1}, E\rangle\\
\mid\ & \texttt{not}\ \ E \mid E \texttt{ == } t \mid v \texttt{ == } E \mid E\ \ \texttt{op}\ \ t \mid n\ \ \texttt{op}\ \ E\\
\mid\ & E\ \ \texttt{bop}\ \ t \mid b\ \ \texttt{bop}\ \ E\\
\mid\ & \texttt{const\_gen}\ E\\
\mid\ & \texttt{shift\_s}\ \ n\ E \mid \texttt{shift\_t}\ \ n\ E \mid \texttt{up\_1d\_s}\ \ n\ E \mid \texttt{up\_1d\_t}\ \ n\ E\\
\mid\ & \texttt{select\_1d\_s}\ \ n\ E \mid \texttt{select\_1d\_t}\ \ n\ E\\
\mid\ & \texttt{map\_s}\ \ E\ t \mid \texttt{map\_s}\ \ v\ E \mid \texttt{map\_t}\ \ E\ t \mid \texttt{map\_t}\ \ v\ E\\
\mid\ & \texttt{map2\_s}\ \ E\ t\ t \mid \texttt{map2\_s}\ \ v\ E\ t \mid \texttt{map2\_s}\ \ v\ v\ E\\
\mid\ & \texttt{map2\_t}\ \ E\ t\ t \mid \texttt{map2\_t}\ \ v\ E\ t \mid \texttt{map2\_t}\ \ v\ v\ E\\
\mid\ & \texttt{reduce\_s}\ \ E\ t \mid \texttt{reduce\_s}\ \ v\ E\\
\mid\ & \texttt{reduce\_t}\ \ E\ t \mid \texttt{reduce\_t}\ \ v\ E\\
\mid\ & \texttt{reshape}\ \ \sigma\ \sigma\ E
\end{aligned}
$$

$$\Gamma \vdash t : \tau$$

$$\Gamma \vdash \texttt{undef} : \tau \qquad \Gamma \vdash n : \mathbb{N} \qquad \Gamma \vdash b : \mathbb{B}$$

$$\frac{\Gamma; (x, \tau) \vdash t : \tau'}{\Gamma \vdash \lambda x : \tau . t : \tau \to \tau'} \qquad \frac{\Gamma \vdash t_1 : \tau \to \tau' \qquad \Gamma \vdash t_2 : \tau}{\Gamma \vdash t_1 t_2 : \tau'} \qquad \frac{(x, \tau) \in \Gamma}{\Gamma \vdash x : \tau}$$

$$\frac{\forall i \in 1 \ldots n \,.\, \Gamma \vdash t_i : \tau}{\Gamma \vdash [t_1, \ldots, t_n] : \texttt{seq}\ n\ \sigma} \qquad \frac{\forall i \in 1 \ldots n \,.\, \Gamma \vdash t_i : \tau_i}{\Gamma \vdash \langle t_1, \ldots, t_n \rangle : \tau_1 \times \ldots \times \tau_n} \qquad \frac{\Gamma \vdash t : \tau_1 \times \ldots \times \tau_n}{\Gamma \vdash t.i : \tau_i}$$

$$\frac{\Gamma \vdash t : \tau \times \ldots \times \tau}{\Gamma \vdash \texttt{tuple\_to\_seq}\ t : \texttt{seq}\ n\ \sigma} \qquad \frac{\Gamma \vdash t : \texttt{seq}\ n\ \sigma}{\Gamma \vdash \texttt{seq\_to\_tuple}\ t : \tau \times \ldots \times \tau}$$

$$\frac{\Gamma \vdash t : \mathbb{B}}{\Gamma \vdash \texttt{not}\ \ t : \mathbb{B}} \qquad \frac{\Gamma \vdash t_1 : \mathbb{N} \qquad \Gamma \vdash t_2 : \mathbb{N}}{\Gamma \vdash t_1\ \ \texttt{==}\ \ t_2 : \mathbb{B}}$$

$$\frac{\Gamma \vdash t_1 : \mathbb{N} \qquad \Gamma \vdash t_2 : \mathbb{N}}{\Gamma \vdash t_1\ \ \texttt{op}\ \ t_2 : \mathbb{N}} \qquad \frac{\Gamma \vdash t_1 : \mathbb{B} \qquad \Gamma \vdash t_2 : \mathbb{B}}{\Gamma \vdash t_1\ \ \texttt{bop}\ \ t_2 : \mathbb{B}}$$

$$\frac{\Gamma \vdash t : \tau}{\Gamma \vdash \texttt{const\_gen}\ t : \tau} \qquad \frac{\Gamma \vdash t_1 : \mathcal{N} \qquad \Gamma \vdash t_2 : \texttt{seq}\ n\ \sigma}{\Gamma \vdash \texttt{shift}\ t_1\ t_2 : \texttt{seq}\ n\ \sigma}$$

$$\frac{\Gamma \vdash t_1 : \mathcal{N} \qquad \Gamma \vdash t_2 : \texttt{seq}\ 1\ \sigma}{\Gamma \vdash \texttt{up\_1d}\ t_1\ t_2 : \texttt{seq}\ n\ \sigma} \qquad \frac{\Gamma \vdash t_1 : \mathcal{N} \qquad \Gamma \vdash t_2 : \texttt{seq}\ n\ \sigma}{\Gamma \vdash \texttt{select\_1d}\ t_1\ t_2 : \texttt{seq}\ 1\ \sigma}$$

$$\frac{\Gamma \vdash t_1 : \mathcal{N} \qquad \Gamma \vdash t_2 : \mathcal{N} \qquad \Gamma \vdash t_3 : \texttt{seq}\ n\ \sigma}{\Gamma \vdash \texttt{partition}\ t_1\ t_2\ t_3 : \texttt{seq}\ n'\ (\texttt{seq}\ n''\ \sigma)} \qquad \frac{\Gamma \vdash t : \texttt{seq}\ n'\ (\texttt{seq}\ n''\ \sigma)}{\Gamma \vdash \texttt{unpartition}\ t : \texttt{seq}\ n\ \sigma}$$

$$\frac{\Gamma \vdash t_1 : \sigma \to \sigma' \qquad \Gamma \vdash t_2 : \texttt{seq}\ n\ \sigma}{\Gamma \vdash \texttt{map}\ t_1\ t_2 : \texttt{seq}\ n\ \sigma'} \qquad \frac{\Gamma \vdash t_1 : \sigma \to \sigma' \to \sigma'' \qquad \Gamma \vdash t_2 : \texttt{seq}\ n\ \sigma \qquad \Gamma \vdash t_3 : \texttt{seq}\ n\ \sigma'}{\Gamma \vdash \texttt{map2}\ t_1\ t_2\ t_3 : \texttt{seq}\ n\ \sigma''}$$

$$\frac{\Gamma \vdash t_1 : (\sigma \times \sigma) \to \sigma \qquad \Gamma \vdash t_2 : \texttt{seq}\ n\ \sigma}{\Gamma \vdash \texttt{reduce}\ t_1\ t_2 : \texttt{seq}\ 1\ \sigma'}$$

Figure 1. $\mathsf{L}^{\mathsf{seq}}$ Typing Rules

$$(\lambda x : \tau.\ t) v \rightsquigarrow^p t[v/x] \qquad \langle v_1, \ldots, v_n \rangle.i \rightsquigarrow^p v_i$$

$$\texttt{tuple\_to\_seq}\ \langle v_1, \ldots, v_n \rangle \rightsquigarrow^p [v_1, \ldots, v_n] \qquad \texttt{seq\_to\_tuple}\ [v_1, \ldots, v_n] \rightsquigarrow^p \langle v_1, \ldots, v_n \rangle$$

$$\texttt{not True} \rightsquigarrow^p \texttt{False} \qquad \texttt{not False} \rightsquigarrow^p \texttt{True} \qquad n\ \ \texttt{==}\ \ n' \rightsquigarrow^p n\ [\![\texttt{==}]\!]\ n'$$

$$n\ op\ n' \rightsquigarrow^p n\ [\![op]\!]\ n' \qquad b\ bop\ b' \rightsquigarrow^p b\ [\![bop]\!]\ b'$$

$$\texttt{const\_gen}\ v \rightsquigarrow^p v$$

$$\texttt{shift}\ n'\ [v_1, \ldots, v_n] \rightsquigarrow^p [undef, \ldots, v_1, \ldots, v_{n-n'}]$$

$$\texttt{up\_1d}\ n\ [v] \rightsquigarrow^p \overbrace{[v, \ldots, v]}^{n} \qquad \texttt{select\_1d}\ n'\ [v_1, \ldots, v_n] \rightsquigarrow^p [v_{n'}]$$

$$\texttt{partition}\ no\ ni\ [v_1, \ldots, v_n] \rightsquigarrow^p [[v_1, \ldots, v_{ni}], \ldots, [v_{n-ni+1}, \ldots, v_n]]$$

$$\texttt{unpartition}\ [[v_1, \ldots, v_{ni}], \ldots, [v_{n-ni+1}, \ldots, v_n]] \rightsquigarrow^p [v_1, \ldots, v_n]$$

$$\texttt{map}\ (\lambda x : \tau.\ t)\ [v_1, \ldots, v_n] \rightsquigarrow^p [(\lambda x_1 : \tau.\ t)\ v_1, \ldots, (\lambda x_n : \tau.\ t)\ v_n]$$

$$\texttt{map2}\ (\lambda x : \tau.\ (\lambda x' : \tau'.\ t))\ [v_1, \ldots, v_n]\ [v'_1, \ldots, v'_n] \rightsquigarrow^p [(\lambda x_1 : \tau.\ (\lambda x'_1 : \tau'.\ t))\ v_1\ v'_1, \ldots, (\lambda x_n : \tau.\ (\lambda x'_n : \tau.\ t))\ v_n\ v'_n]$$

$$\texttt{reduce}\ (\lambda x : \tau \times \tau.\ t)\ [v_1, \ldots, v_n] \rightsquigarrow^p [(\lambda x_1 : \tau \times \tau.\ t)\ v_1\ \langle(\lambda x_2 : \tau \times \tau.\ t)\ v_2\ \langle \ldots ((\lambda x_n : \tau \times \tau.\ t)\langle v_1, v_2 \rangle)\rangle\rangle]$$

Figure 2. $\mathsf{L}^{\mathsf{seq}}$ Primitive Reduction Rules

$$\Gamma \vdash \mathtt{t} : \tau$$

$$\frac{\forall i \in 1 \ldots n \,.\, \Gamma \vdash t_i : \tau}{\Gamma \vdash [t_1, \ldots, t_n]_s : \mathtt{sseq}\ v\ \sigma} \qquad \frac{\forall i \in 1 \ldots n \,.\, \Gamma \vdash t_i : \tau}{\Gamma \vdash [t_1, \ldots, t_n]_t : \mathtt{tseq}\ v\ i\ \sigma}$$

$$\frac{\forall i \in 1 \ldots n \,.\, \Gamma \vdash t_i : \tau_i}{\Gamma \vdash \langle t_1, \ldots, t_n \rangle : \tau_1 \times \ldots \times \tau_n} \qquad \frac{\Gamma \vdash t : \tau_1 \times \ldots \times \tau_n}{\Gamma \vdash t.i : \tau_i}$$

$$\frac{\Gamma \vdash t_1 : \mathcal{N} \qquad \Gamma \vdash t_2 : \mathtt{sseq}\ v\ \sigma}{\Gamma \vdash \mathtt{shift\_s}\ t_1\ t_2 : \mathtt{sseq}\ v\ \sigma} \qquad \frac{\Gamma \vdash t_1 : \mathcal{N} \qquad \Gamma \vdash t_2 : \mathtt{tseq}\ v\ i\ \sigma}{\Gamma \vdash \mathtt{shift\_t}\ t_1\ t_2 : \mathtt{tseq}\ v\ i\ \sigma}$$

$$\frac{\Gamma \vdash t_1 : \mathcal{N} \qquad \Gamma \vdash t_2 : \mathtt{sseq}\ 1\ \sigma}{\Gamma \vdash \mathtt{up\_1d\_s}\ t_1\ t_2 : \mathtt{sseq}\ v\ \sigma} \qquad \frac{\Gamma \vdash t_1 : \mathcal{N} \qquad \Gamma \vdash t_2 : \mathtt{tseq}\ 1\ i\ \sigma}{\Gamma \vdash \mathtt{up\_1d\_t}\ t_1\ t_2 : \mathtt{sseq}\ v\ i'\ \sigma}$$

$$\frac{\Gamma \vdash t_1 : \mathcal{N} \qquad \Gamma \vdash t_2 : \mathtt{sseq}\ v\ \sigma}{\Gamma \vdash \mathtt{select\_1d\_s}\ t_1\ t_2 : \mathtt{sseq}\ 1\ \sigma} \qquad \frac{\Gamma \vdash t_1 : \mathcal{N} \qquad \Gamma \vdash t_2 : \mathtt{tseq}\ v\ i\ \sigma}{\Gamma \vdash \mathtt{select\_1d\_t}\ t_1\ t_3 : \mathtt{tsseq}\ 1\ i'\ \sigma}$$

$$\frac{\Gamma \vdash t_1 : \sigma \rightarrow \sigma' \qquad \Gamma \vdash t_2 : \mathtt{sseq}\ v\ \sigma}{\Gamma \vdash \mathtt{map\_s}\ t_1\ t_2 : \mathtt{sseq}\ v\ \sigma'} \qquad \frac{\Gamma \vdash t_1 : \sigma \rightarrow \sigma' \qquad \Gamma \vdash t_2 : \mathtt{tseq}\ v\ i\ \sigma}{\Gamma \vdash \mathtt{map\_t}\ t_1\ t_2 : \mathtt{tseq}\ v\ i\ \sigma'}$$

$$\frac{\Gamma \vdash t_1 : \sigma \rightarrow \sigma' \rightarrow \sigma'' \qquad \Gamma \vdash t_2 : \mathtt{sseq}\ v\ \sigma \qquad \Gamma \vdash t_3 : \mathtt{sseq}\ v\ \sigma'}{\Gamma \vdash \mathtt{map2\_s}\ t_1\ t_2\ t_3 : \mathtt{sseq}\ v\ \sigma''}$$

$$\frac{\Gamma \vdash t_1 : \sigma \rightarrow \sigma' \rightarrow \sigma'' \qquad \Gamma \vdash t_2 : \mathtt{tseq}\ v\ i\ \sigma \qquad \Gamma \vdash t_3 : \mathtt{tseq}\ v\ i\ \sigma'}{\Gamma \vdash \mathtt{map2\_t}\ t_1\ t_2\ t_3 : \mathtt{tseq}\ v\ i\ \sigma''}$$

$$\frac{\Gamma \vdash t_1 : (\sigma \times \sigma) \rightarrow \sigma \qquad \Gamma \vdash t_2 : \mathtt{sseq}\ v\ \sigma}{\Gamma \vdash \mathtt{reduce\_s}\ t_1\ t_2 : \mathtt{sseq}\ 1\ \sigma'} \qquad \frac{\Gamma \vdash t_1 : (\sigma \times \sigma) \rightarrow \sigma \qquad \Gamma \vdash t_2 : \mathtt{tseq}\ v\ i\ \sigma}{\Gamma \vdash \mathtt{reduce\_t}\ t_1\ t_2 : \mathtt{tseq}\ 1\ i'\ \sigma'}$$

$$\frac{\Gamma \vdash t : \sigma}{\Gamma \vdash \mathtt{reshape}\ \sigma\ \sigma'\ t : \sigma'}$$

Figure 3. L[st] typing rules. Duplicates from the L[seq] are omitted.

$$\mathtt{shift\_s}\ n'\ [v_1, \ldots, v_n] \rightsquigarrow^p [undef, \ldots, v_1, \ldots, v_{n-n'}]_t$$

$$\mathtt{shift\_t}\ n'\ [v_1, \ldots, v_n, \ldots, v_{n+i}]_t \rightsquigarrow^p [undef, \ldots, v_1, \ldots, v_{n-n'}, \ldots, v_{n+i}]_t$$

$$\mathtt{up\_1d\_s}\ n\ [v]_s \rightsquigarrow^p \overbrace{[v, \ldots, v]_s}^{n} \qquad \mathtt{up\_1d\_t}\ n\ [v, \ldots, v_{n+i}]_t \rightsquigarrow^p [\overbrace{v, \ldots, v}^{n}, \ldots, v_{n+i}]_t$$

$$\mathtt{select\_1d\_s}\ n'\ [v_1, \ldots, v_n]_s \rightsquigarrow^p [v_{n'}]_s \qquad \mathtt{select\_1d\_t}\ n'\ [v_1, \ldots, v_n, \ldots, v_{n+i}]_t \rightsquigarrow^p [v_{n'}, \ldots, v_{n+i}]_t$$

$$\mathtt{map\_s}\ (\lambda x : \tau.\ t)\ [v_1, \ldots, v_n]_s \rightsquigarrow^p [(\lambda x_1 : \tau.\ t)\ v_1, \ldots, (\lambda x_n : \tau.\ t)\ v_n]_s$$

$$\mathtt{map\_t}\ (\lambda x : \tau.\ t)\ [v_1, \ldots, v_n, \ldots, v_{n+i}]_t \rightsquigarrow^p [(\lambda x_1 : \tau.\ t)\ v_1, \ldots, (\lambda x_n : \tau.\ t)\ v_n, \ldots, v_{n+i}]_t$$

$$\mathtt{map2\_s}\ (\lambda x : \tau.\ (\lambda x' : \tau'.\ t))\ [v_1, \ldots, v_n]_s\ [v'_1, \ldots, v'_n]_s \rightsquigarrow^p [(\lambda x_1 : \tau.\ (\lambda x'_1 : \tau'.\ t))\ v_1\ v'_1, \ldots, (\lambda x_n : \tau.\ (\lambda x'_n : \tau'.\ t))\ v_n\ v'_n]_s$$

$$\mathtt{map2\_t}\ (\lambda x : \tau.\ (\lambda x' : \tau.\ t))\ [v_1, \ldots, v_n, \ldots, v_{n+i}]_t\ [v'_1, \ldots, v'_n, \ldots, v'_{n+i}]_t \rightsquigarrow^p$$
$$[(\lambda x_1 : \tau.\ (\lambda x'_1 : \tau.\ t))\ v_1\ v'_1, \ldots, (\lambda x_n : \tau.\ (\lambda x'_n : \tau.\ t))\ v_n\ v'_n\ \ \ldots v_{n+i}, v'_{n+i}]_t$$

$$\mathtt{reduce\_s}\ (\lambda x : \tau \times \tau.\ t)\ [v_1, \ldots, v_n]_s \rightsquigarrow^p [(\lambda x_1 : \tau \times \tau.\ t)\ v_1\ \langle(\lambda x_2 : \tau \times \tau.\ t)\ v_2\ \langle \ldots ((\lambda x_n : \tau \times \tau.\ t)\langle v_1, v_2\rangle)\rangle\rangle]_s$$

$$\mathtt{reduce\_t}\ (\lambda x : \tau \times \tau.\ t)\ [v_1, \ldots, v_n, \ldots, v_{n+i}]_t \rightsquigarrow^p [(\lambda x_1 : \tau \times \tau.\ t)\ v_1\ \langle(\lambda x_2 : \tau \times \tau.\ t)\ v_2\ \langle \ldots ((\lambda x_n : \tau \times \tau.\ t)\langle v_1, v_2\rangle)\rangle\rangle, \ldots, v_{n+i}]_t$$

$\mathtt{reshape}\ \sigma\ \sigma'\ v \rightsquigarrow^p v'$ s.t. v' and v are equal when converted to a flat seq

Figure 4. L[st] Primitive Reduction Rules. Duplicates from the L[seq] are omitted.

C.5 L^{st} Evaluation Contexts

$$
\begin{array}{rl}
\mathfrak{c} ::= & [\cdot] \mid \mathfrak{c}\ t \mid (\lambda x : \tau.\ t)\ \mathfrak{c} \\
\mid & [\mathfrak{c}, \ldots, t_n]_s \mid [t_1, \ldots, \mathfrak{c}, \ldots, t_n]_s \mid [t_1, \ldots, t_{n-1}, \mathfrak{c}]_s \mid [\mathfrak{c}, \ldots, t_n]_t \\
\mid & [t_1, \ldots, \mathfrak{c}, \ldots, t_n]_t \mid [t_1, \ldots, t_{n-1}, \mathfrak{c}]_t \\
\mid & \langle \mathfrak{c}, \ldots, t_n \rangle \mid \langle t_1, \ldots, \mathfrak{c}, \ldots, t_n \rangle \mid \langle t_1, \ldots, t_{n-1}, \mathfrak{c} \rangle \\
\mid & \texttt{not}\ \mathfrak{c} \mid \mathfrak{c} == t \mid t == \mathfrak{c} \mid \mathfrak{c}\ \texttt{op}\ t \mid n\ \texttt{op}\ \mathfrak{c} \\
\mid & \mathfrak{c}\ \texttt{bop}\ t \mid b\ \texttt{bop}\ \mathfrak{c} \\
\mid & \texttt{const\_gen}\ \mathfrak{c} \\
\mid & \texttt{shift\_s}\ n\ \mathfrak{c} \mid \texttt{shift\_t}\ n\ \mathfrak{c} \mid \texttt{up\_1d\_s}\ n\ \mathfrak{c} \mid \texttt{up\_1d\_t}\ n\ \mathfrak{c} \\
\mid & \texttt{select\_1d\_s}\ n\ \mathfrak{c} \mid \texttt{select\_1d\_t}\ n\ \mathfrak{c} \\
\mid & \texttt{map\_s}\ \mathfrak{c}\ t \mid \texttt{map\_s}\ t\ \mathfrak{c} \mid \texttt{map\_t}\ \mathfrak{c}\ t \mid \texttt{map\_t}\ t\ \mathfrak{c} \\
\mid & \texttt{map2\_s}\ \mathfrak{c}\ t\ t \mid \texttt{map2\_s}\ t\ \mathfrak{c}\ t \mid \texttt{map2\_s}\ t\ t\ \mathfrak{c} \\
\mid & \texttt{map2\_t}\ \mathfrak{c}\ t\ t \mid \texttt{map2\_t}\ t\ \mathfrak{c}\ t \mid \texttt{map2\_t}\ t\ t\ \mathfrak{c} \\
\mid & \texttt{reduce\_s}\ \mathfrak{c}\ t \mid \texttt{reduce\_s}\ t\ \mathfrak{c} \\
\mid & \texttt{reduce\_t}\ \mathfrak{c}\ t \mid \texttt{reduce\_t}\ t\ \mathfrak{c} \\
\mid & \texttt{reshape}\ \sigma\ \sigma\ E
\end{array}
$$

References

[1] Thaddeus Koehn and Peter Athanas. 2016. Arbitrary streaming permutations with minimum memory and latency. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 1–6.

[2] Xilinx, Inc. 2018. *Zynq-7000 SoC Data Sheet: Overview*. Xilinx, Inc.