VideoCore IV Typed Assembly Language

Version 0.1

El Pin Al

Index

1. Syntax

To define the syntax, there are some primitive terms:

- i denotes an integer.
- $i4$ denotes a 4-bit integer in range $[0, 15]$.
- $i4^{*}$ denotes a 4-bit integer in range $[1, 16]$.
- $i7^{*}$ denotes a 7-bit integer in range $[1, 128]$.
- f denotes a floating-point number.

- l denotes a label.
- nat denotes an integer which is greater than or equal to 0.
- In general, ε denotes an empty construct.
- α denotes a variable.

The syntax is given below.

$$
\begin{array}{lll}
\nu ::= & n \mid r & \text{operands} \\
n ::= & i \mid f & \text{numbers} \\
r ::= & rr \mid wr & \text{registers} \\
rr ::= & rwr \mid \text{uniform} \mid \text{element\_number} \mid \text{vpmr} & \text{readable registers} \\
urr ::= & a \mid \text{element\_number} & \text{unconstrained readable registers} \\
wr ::= & rwr \mid \text{uniforms\_address} & \text{writable registers} \\
 & \mid tmu \mid \text{broadcast} \mid \text{vpmw} & \\
rwr ::= & ga \mid rf & \text{both readable and writable registers} \\
a ::= & ga \mid sa & \text{accumulators} \\
ga ::= & \text{r0} \mid \text{r1} \mid \text{r2} \mid \text{r3} & \text{general-purpose accumulators} \\
sa ::= & \text{r4} \mid \text{r5} & \text{special-purpose accumulators} \\
rf ::= & A \mid B & \text{register files} \\
A ::= & \text{ra0} \mid \ldots \mid \text{ra31} & \text{locations in register file A} \\
B ::= & \text{rb0} \mid \ldots \mid \text{rb31} & \text{locations in register file B} \\
tmu ::= & \text{tmu0} \mid \text{tmu1} & \text{TMU registers} \\
Y ::= & \text{y}_0 \mid \ldots \mid \text{y}_{63} & \text{VPM Y} \\
\iota ::= & \text{rotate}(wr, rr, i4) \mid \text{mov}(wr, \nu) & \text{instructions} \\
 & \mid \text{setup\_vpm\_read}\left(Y, i4^{*}\right) & \\
 & \mid \text{setup\_vpm\_write}(Y) & \\
 & \mid \text{setup\_dma\_load}\left(Y, i4^{*}\right) \mid \text{start\_dma\_load}(rr) & \\
 & \mid \text{wait\_dma\_load} & \\
 & \mid \text{setup\_dma\_store}\left(Y, i7^{*}\right) \mid \text{start\_dma\_store}(rr) & \\
 & \mid \text{wait\_dma\_store} &
\end{array}
$$

$$
\begin{array}{lll}
cc & ::= \mathrm{Z} \mid \mathrm{N} \mid \mathrm{C} & \text{condition classes} \\
c & ::= \mathrm{set}(cc) \mid \mathrm{clear}(cc) & \text{conditions} \\
qc & ::= \mathrm{all}(c) \mid \mathrm{any}(c) & \text{quantified conditions} \\
s & ::= \varepsilon \mid \mathrm{load}\langle tmu \rangle & \text{signals} \\
ci & ::= (\iota,\varepsilon) \mid (\iota,c) & \text{conditional instructions} \\
csi & ::= (ci,s) & \text{conditional instructions with signals} \\
I & ::= \mathrm{jmp}(l)\ ;\ csi\ ;\ csi\ ;\ csi & \text{instruction sequences} \\
 & \mid (ci,\mathrm{thread\_end})\ ;\ csi\ ;\ csi & \\
 & \mid csi\ ;\ I & \\
 & \mid \mathrm{if}\ qc\ \mathrm{jmp}(l)\ ;\ csi\ ;\ csi\ ;\ csi\ ;\ I &
\end{array}
$$

Abstract machine:

$$
\begin{array}{lll}
R & ::= \varepsilon \mid R[rr \mapsto arr] & \text{register contexts} \\
arr & ::= [n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n] & \text{arrays} \\
U & ::= \varepsilon \mid n \circ U & \text{uniforms} \\
T & ::= \varepsilon \mid arr \mid arr :: arr \mid arr :: arr :: arr & \text{TMU} \\
 & \mid arr :: arr :: arr :: arr & \\
V & ::= \varepsilon \mid V[Y \mapsto at] & \text{VPM states} \\
P & ::= \varepsilon \mid P[l \mapsto I] & \text{programs} \\
vpmq & ::= \varepsilon \mid (Y,i) :: vpmq & \text{VPM read queues} \\
M & ::= (R,U,T,V,vpmq,P,I) & \text{machines}
\end{array}
$$

2. Type system

The type syntax is defined as follows:

τ ::=	$b \mid at \mid vt \mid \Psi$	types
p ::=	$nat \mid \alpha \mid p+p \mid nat \times p$	pointers
b ::=	$\mathrm{int}(i) \mid \mathrm{int}(?) \mid \mathrm{float} \mid \mathrm{ptr}(p) \mid \mathrm{code}(\Theta)$	basic types
at ::=	$[b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b]$	array types
vt ::=	$\mathrm{vec}(b,\alpha)$	vector types
Γ ::=	$\varepsilon \mid \Gamma[rr \mapsto at]$	register context types
Ψ ::=	$\varepsilon \mid b \circ \Psi$	uniforms types
Σ ::=	$\varepsilon \mid at \mid at :: at \mid at :: at :: at \mid at :: at :: at :: at$	TMU types
Π ::=	$\varepsilon \mid \Pi[Y \mapsto at]$	VPM state types
$vpmw\_addr$ ::=	$\varepsilon \mid Y$	VPM write addresses
C ::=	$(vpmq, \Pi, vpmw\_addr)$	VPM compound types
u ::=	$\mathrm{u0} \mid \mathrm{u1} \mid \mathrm{u2}$	uniform-access countdowns
Θ ::=	$(\Gamma, \Psi, u, \Sigma, C, \Omega, wr)$	state types
Φ ::=	$\varepsilon \mid \Phi[l \mapsto \Theta]$	program types
DLS ::=	$\varepsilon \mid \left(Y, i4^{*}\right)$	DMA load setups
DSS ::=	$\varepsilon \mid \left(Y, i7^{*}\right)$	DMA store setups
DL ::=	$\varepsilon \mid \left(Y, i4^{*}, \tau\right)$	DMA loads
DS ::=	$\varepsilon \mid \left(Y, i7^{*}, \tau\right) \mid \mathrm{type\_preserving}$	DMA stores

2.1. Propositions

$p_1 \,..\, p_2$ represents a range $[p_1, p_2)$.

$\varphi ::= p \,..\, p \mid \varphi \vee \varphi$ propositions

2.2. Memory representaion

$\Omega ::= \{\Xi \mid \varphi\}$ memory subset types

$\Xi ::= \varepsilon \mid \Xi[p \mapsto \tau]$ memory types

The evaluation rules of memory subset types and memory types are given below. The equality rules for p are not defined here. There are the abuses of notations of a form $x[y \mapsto z]$.

$$\frac{p \text{ is in } \varphi}{\{\Xi \mid \varphi\}(p) \rightarrow_{\Omega} \Xi(p)}$$

$$\frac{p = p_1}{\Xi\left[p_1 \mapsto \tau\right](p) \rightarrow_{\Xi} \tau}$$

$$\frac{p \neq p_1}{\Xi\left[p_1 \mapsto \tau\right](p) \rightarrow_{\Xi} \Xi(p)}$$

2.3. Typing rules

Typing rules are defined as follows. Note that $dom(\Gamma)$ represents the domain of a context Γ, a map from read / write registers to array types.

numbers:

$$\vdash i : \mathrm{int}(i) \qquad \vdash f : \mathrm{float}$$

subtype relations:

$$\mathrm{int}(i) <: \mathrm{int}(?) \qquad \mathrm{int}(?) <: \mathrm{int}(?) \qquad \frac{\forall i \;.\; at_1[i] <: at_2[i]}{at_1 <: at_2}\text{(S-Array)}$$

$$\frac{m \leq n}{\mathrm{vec}(b, m) <: \mathrm{vec}(b, n)}\text{(S-Vector)}$$

$$\Gamma <: \Gamma$$

$$\frac{\Gamma_1 <: \Gamma_2 \qquad rr \notin dom(\Gamma_1)}{\Gamma_1[rr \mapsto at] <: \Gamma_2}\text{(S-Ctx-Width)} \qquad \frac{\Gamma_1 <: \Gamma_2 \qquad at_1 <: at_2}{\Gamma_1[rr \mapsto at_1] <: \Gamma_2[rr \mapsto at_2]}\text{(S-Ctx-Depth)}$$

pointers:

$$\Xi \vdash p : \Xi(p)$$

well-formed memory subsets:

$$\frac{\forall p \in dom(\Xi) \;.\; p \,..\, \big(p + size\_of\big(\Xi(p)\big)\big) \text{ is in } \varphi}{\vdash \{\Xi \mid \varphi\}}$$

registers:

$$\Gamma \vdash rr : \Gamma(rr) \qquad \frac{\Psi_1 \equiv b \circ \Psi_2}{\vdash \text{uniform} : b \mid \Psi_1 \to \Psi_2}$$

$$\vdash \text{element\_number} : [\text{int}(0), \text{int}(1), \ldots, \text{int}(15)]$$

$$\frac{vpmq_1 \equiv \left(vpmq_2 :: (Y, i)\right) \qquad i \geq 2 \qquad \Pi \vdash Y : at}{\Pi \vdash \text{vpmr} : at \mid vpmq_1 \to vpmq_2 :: \left(inc(Y), i-1\right)}$$

$$\frac{vpmq_1 \equiv \left(vpmq_2 :: (Y, 1)\right) \qquad \Pi \vdash Y : at}{\Pi \vdash \text{vpmr} : at \mid vpmq_1 \to vpmq_2}$$

VPM:

$$\Pi \vdash Y : \Pi(Y)$$

$$\frac{\Pi \vdash Y : at \qquad at <: vt}{\Pi \vdash (Y, 1) : vt}$$

$$\frac{\Pi \vdash Y : at \qquad at <: vt_1 \qquad n \leq 62 \qquad i7^* \geq 2 \qquad \Pi \vdash \left(\mathrm{y}_{n+1}, i7^* - 1\right) : vt_2}{\Pi \vdash \left(\mathrm{y}_n, i7^*\right) : concat(vt_1, vt_2)}$$

instructions:

$$\frac{rr \not\equiv wr_{before} \qquad at \equiv rotate(\Gamma(rr), i4)}{wr_{before} \vdash \text{rotate}(rwr, rr, i4) : \Gamma \to \Gamma[rwr \mapsto at] \ ; \ rwr}$$

$$\frac{\vdash \text{uniform} : b \mid \Psi_1 \to \Psi_2 \qquad at \equiv array(b)}{\text{u0} \vdash \text{rotate}(rwr, \text{uniform}, i4) : \Gamma \to \Gamma[rwr \mapsto at] \mid \Psi_1 \to \Psi_2 \ ; \ rwr}$$

$$\frac{\vdash \text{element\_number} : at}{\vdash \text{rotate}(rwr, \text{element\_number}, i4) : \Gamma \to \Gamma\left[rwr \mapsto rotate(at, i4)\right] \ ; \ rwr}$$

$$\frac{\Pi \vdash \text{vpmr} : at \mid vpmq_1 \to vpmq_2}{\Pi \vdash \text{rotate}(rwr, \text{vpmr}, i4) : vpmq_1 \to vpmq_2 \mid \Gamma \to \Gamma\left[rwr \mapsto rotate(at, i4)\right] \ ; \ rwr}$$

$$\frac{rr \not\equiv wr_{before} \qquad \Gamma \vdash rr : at \qquad fst\big(rotate(at, i4)\big) \equiv b}{\Gamma;\, wr_{before} \vdash \mathrm{rotate}\big(\mathrm{broadcast}, rr, i4\big) : \Gamma \to \Gamma\big[\mathrm{r5} \mapsto array(b)\big]}$$

$$\frac{\vdash \mathrm{uniform} : b \mid \Psi_1 \to \Psi_2}{\vdash \mathrm{rotate}\big(\mathrm{broadcast}, \mathrm{uniform}, i4\big) : \Gamma \to \Gamma\big[\mathrm{r5} \mapsto array(b)\big] \mid \Psi_1 \to \Psi_2}$$

$$\frac{\vdash \mathrm{element\_number} : at \qquad b \equiv fst\big(rotate(at, i4)\big)}{\vdash \mathrm{rotate}\big(\mathrm{broadcast}, \mathrm{element\_number}, i4\big) : \Gamma \to \Gamma\big[\mathrm{r5} \mapsto array(b)\big]}$$

$$\frac{\Pi \vdash \mathrm{vpmr} : at \mid vpmq_1 \to vpmq_2 \qquad fst\big(rotate(at, i4)\big) \equiv b}{\Pi \vdash \mathrm{rotate}\big(\mathrm{broadcast}, \mathrm{vpmr}, i4\big) : \Gamma \to \Gamma\big[\mathrm{r5} \mapsto array(b)\big] \mid vpmq_1 \to vpmq_2}$$

$$\frac{rr \not\equiv wr_{before} \qquad \Gamma \vdash rr : at \qquad fst\big(rotate(at, i4)\big) \equiv \mathrm{ptr}(p) \qquad \Xi \vdash p : \Psi_2}{\Gamma;\, \Xi\,;\, wr_{before} \vdash \mathrm{rotate}\big(\mathrm{uniforms\_address}, rr, i4\big) : \Psi_1 \to \Psi_2 \ \mid u \to \mathrm{u2}}$$

$$\frac{\vdash \mathrm{uniform} : \mathrm{ptr}(p) \mid \Psi_1 \to \Psi_2 \qquad \Xi \vdash p : \Psi_3}{\Xi \vdash \mathrm{rotate}\big(\mathrm{uniforms\_address}, \mathrm{uniform}, i4\big) : \Psi_1 \to \Psi_3 \ \mid \mathrm{u0} \to \mathrm{u2}}$$

$$\frac{\Pi \vdash \mathrm{vpmr} : at \mid vpmq_1 \to vpmq_2 \qquad fst\big(rotate(at, i4)\big) \equiv \mathrm{ptr}(p) \qquad \Xi \vdash p : \Psi_2}{\Pi;\, \Xi \vdash \mathrm{rotate}\big(\mathrm{uniforms\_address}, \mathrm{vpmr}, i4\big) : \Psi_1 \to \Psi_2 \mid vpmq_1 \to vpmq_2 \mid u \to \mathrm{u2}}$$

$$\frac{rr \not\equiv wr_{before} \qquad \Gamma \vdash rr : at \qquad notfull\big(\Sigma^{tmu}\big)}{\Gamma;\ \Xi\,;\, wr_{before} \vdash \mathrm{rotate}\big(tmu, rr, i4\big) : \Sigma^{tmu} \to rotate\Big(map\Big(unwrap_{\Xi}, at\Big), i4\Big) :: \Sigma^{tmu}}$$

$$\frac{\vdash \mathrm{uniform} : \mathrm{ptr}(p) \ \mid \Psi_1 \to \Psi_2 \qquad \Xi \vdash p : b \qquad notfull\big(\Sigma^{tmu}\big)}{\mathrm{u0};\, \Xi \vdash \mathrm{rotate}\big(tmu, \mathrm{uniform}, i4\big) : \Sigma^{tmu} \to array(b) :: \Sigma^{tmu} \ \mid \Psi_1 \to \Psi_2}$$

$$\frac{\Pi \vdash \mathrm{vpmr} : at \mid vpmq_1 \to vpmq_2 \qquad notfull\big(\Sigma^{tmu}\big)}{\Pi;\, \Xi \vdash \mathrm{rotate}\big(tmu, \mathrm{vpmr}, i4\big) : \Sigma^{tmu} \to rotate\Big(map\Big(unwrap_{\Xi}, at\Big), i4\Big) :: \Sigma^{tmu}}$$

$$\frac{rr \not\equiv wr_{before} \qquad at \equiv rotate\big(\Gamma(rr), i4\big)}{wr_{before} \vdash \mathrm{rotate}\big(\mathrm{vpmw}, rr, i4\big) : \Pi \to \Pi[Y \mapsto at] \mid Y \to inc(Y)}$$

$$\frac{\vdash \mathrm{uniform} : b \ \mid\ \Psi_1 \to \Psi_2 \qquad at \equiv array(b)}{\mathrm{u0} \vdash \mathrm{rotate}\big(\mathrm{vpmw}, \mathrm{uniform}, i4\big) : \Pi \to \Pi[Y \mapsto at] \mid \Psi_1 \to \Psi_2 \mid Y \to inc(Y)}$$

$$\frac{\vdash \text{element\_number} : at}{\vdash \text{rotate}(\text{vpmw}, \text{element\_number}, i4) : \Pi \to \Pi\big[Y \mapsto rotate(at, i4)\big] \mid Y \to inc(Y)}$$

$$\frac{\vdash n : b}{\vdash \text{mov}(rwr, n) : \Gamma \to \Gamma\big[rwr \mapsto array(b)\big] \ ; rwr}$$

$$\frac{\Gamma \vdash urr : at}{\vdash \text{mov}(rwr, urr) : \Gamma \to \Gamma[rwr \mapsto at] \ ; rwr}$$

$$\frac{\Gamma \vdash rf : at \qquad rf \not\equiv wr_{before}}{wr_{before} \vdash \text{mov}(rwr, rf) : \Gamma \to \Gamma[rwr \mapsto at] \ ; rwr}$$

$$\frac{\vdash \text{uniform} : b \ \mid \ \Psi_1 \to \Psi_2}{\text{u0} \vdash \text{mov}(rwr, \text{uniform}) : \Gamma \to \Gamma\big[rwr \mapsto array(b)\big] \ \mid \Psi_1 \to \Psi_2; rwr}$$

$$\frac{\Pi \vdash \text{vpmr} : at \mid vpmq_1 \to vpmq_2}{\Pi \vdash \text{mov}(rwr, \text{vpmr}) : \Gamma \to \Gamma[rwr \mapsto at] \mid vpmq_1 \to vpmq_2 \ ; rwr}$$

$$\frac{\vdash n : b}{\vdash \text{mov}(\text{broadcast}, n) : \Gamma \to \Gamma\big[\text{r5} \mapsto array(b)\big]}$$

$$\frac{\Gamma \vdash urr : at \qquad fst(at) \equiv b}{\Gamma \vdash \text{mov}(\text{broadcast}, urr) : \Gamma \to \Gamma\big[\text{r5} \mapsto array(b)\big]}$$

$$\frac{\Gamma \vdash rf : at \qquad fst(at) \equiv b \qquad rf \not\equiv wr_{before}}{\Gamma; wr_{before} \vdash \text{mov}(\text{broadcast}, rf) : \Gamma \to \Gamma\big[\text{r5} \mapsto array(b)\big]}$$

$$\frac{\vdash \text{uniform} : b \mid \Psi_1 \to \Psi_2}{\text{u0} \vdash \text{mov}(\text{broadcast}, \text{uniform}) : \Gamma \to \Gamma\big[\text{r5} \mapsto array(b)\big] \mid \Psi_1 \to \Psi_2}$$

$$\frac{\Pi \vdash \text{vpmr} : at \mid vpmq_1 \to vpmq_2}{\Pi \vdash \text{mov}(\text{broadcast}, \text{vpmr}) : \Gamma \to \Gamma[\text{r5} \mapsto at] \mid vpmq_1 \to vpmq_2}$$

$$\frac{\Gamma \vdash a : at \qquad fst(at) \equiv \text{ptr}(p) \qquad \Xi \vdash p : \Psi_2}{\Gamma; \Xi \vdash \text{mov}(\text{uniforms\_address}, a) : \Psi_1 \to \Psi_2 \mid u \to \text{u2}}$$

$$\frac{\Gamma \vdash rf : at \qquad fst(at) \equiv \text{ptr}(p) \qquad \Xi \vdash p : \Psi_2 \qquad rf \not\equiv wr_{before}}{\Gamma; \ \Xi \ ; wr_{before} \vdash \text{mov}(\text{uniforms\_address}, rf) : \Psi_1 \to \Psi_2 \mid u \to \text{u2}}$$

$$\frac{\vdash \mathrm{uniform} : \mathrm{ptr}(p) \mid \Psi_1 \to \Psi_2 \qquad \Xi \vdash p : \Psi_3}{\Xi \vdash \mathrm{mov}(\mathrm{uniforms\_address}, \mathrm{uniform}) : \Psi_1 \to \Psi_3 \mid \mathrm{u0} \to \mathrm{u2}}$$

$$\frac{\Pi \vdash \mathrm{vpmr} : at \mid vpmq_1 \to vpmq_2 \qquad fst(at) \equiv \mathrm{ptr}(p) \qquad \Xi \vdash p : \Psi_2}{\Pi; \Xi \vdash \mathrm{mov}(\mathrm{uniforms\_address}, \mathrm{vpmr}) : vpmq_1 \to vpmq_2 \mid \Psi_1 \to \Psi_2 \mid u \to \mathrm{u2}}$$

$$\frac{\Gamma \vdash a : at \qquad notfull\left(\Sigma^{tmu}\right)}{\Gamma; \Xi \vdash \mathrm{mov}(tmu, a) : \Sigma^{tmu} \to map\left(unwrap_{\Xi}, at\right) :: \Sigma^{tmu}}$$

$$\frac{\Gamma \vdash rf : at \qquad notfull\left(\Sigma^{tmu}\right) \qquad rf \not\equiv wr_{before}}{\Gamma;\ \Xi\, ; wr_{before} \vdash \mathrm{mov}(tmu, rf) : \Sigma^{tmu} \to map\left(unwrap_{\Xi}, at\right) :: \Sigma^{tmu}}$$

$$\frac{\vdash \mathrm{uniform} : \mathrm{ptr}(p) \ \mid\ \Psi_1 \to \Psi_2 \qquad \Xi \vdash p : b \qquad notfull\left(\Sigma^{tmu}\right)}{\mathrm{u0}; \Xi \vdash \mathrm{mov}(tmu, \mathrm{uniform}) : \Sigma^{tmu} \to array(b) :: \Sigma^{tmu} \ \mid \Psi_1 \to \Psi_2}$$

$$\frac{\Pi \vdash \mathrm{vpmr} : at \mid vpmq_1 \to vpmq_2 \qquad notfull\left(\Sigma^{tmu}\right)}{\Pi; \Xi \vdash \mathrm{mov}(tmu, \mathrm{vpmr}) : vpmq_1 \to vpmq_2 \mid \Sigma^{tmu} \to map\left(unwrap_{\Xi}, at\right) :: \Sigma^{tmu}}$$

$$\frac{\vdash n : b}{\vdash \mathrm{mov}(\mathrm{vpmw}, n) : \Pi \to \Pi\left[Y \mapsto array(b)\right] \mid Y \to inc(Y)}$$

$$\frac{\Gamma \vdash urr : at}{\vdash \mathrm{mov}(\mathrm{vpmw}, urr) : \Pi \to \Pi[Y \mapsto at] \mid Y \to inc(Y)}$$

$$\frac{\Gamma \vdash rf : at \qquad rf \not\equiv wr_{before}}{wr_{before} \vdash \mathrm{mov}(\mathrm{vpmw}, rf) : \Pi \to \Pi[Y \mapsto at] \mid Y \to inc(Y)}$$

$$\frac{\vdash \mathrm{uniform} : b \ \mid\ \Psi_1 \to \Psi_2}{\mathrm{u0} \vdash \mathrm{mov}(\mathrm{vpmw}, \mathrm{uniform}) : \Pi \to \Pi\left[Y \mapsto array(b)\right] \mid \Psi_1 \to \Psi_2 \mid Y \to inc(Y)}$$

$$\frac{\Pi \vdash Y : at}{\Pi \vdash \mathrm{setup\_vpm\_read}\left(Y, i4^{*}\right) : vpmq \to \left(Y, i4^{*}\right) :: vpmq}$$

$$\vdash \mathrm{setup\_vpm\_write}(Y) : vpmw\_addr \to Y$$

$$\frac{i + i4^{*} \leq 64}{\vdash \text{setup\_dma\_load}\left(\mathrm{y}_{i}, i4^{*}\right) : DLS \rightarrow \left(\mathrm{y}_{i}, i4^{*}\right)}$$

$$\frac{\Gamma \vdash rr : at \qquad fst(at) \equiv \mathrm{ptr}(p) \qquad \Xi \vdash p : \tau \qquad i4^{*} \times 16 \times 4 \leq size\_of(\tau)}{\left(Y, i4^{*}\right); \Xi \vdash \text{start\_dma\_load}(rr) : \varepsilon \rightarrow \left(Y, i4^{*}, \tau\right)}$$

$$\vdash \text{wait\_dma\_load} : \left(Y, i4^{*}, \tau\right) \rightarrow \varepsilon \mid \Pi \rightarrow \Pi\ dma\_load\left(Y, i4^{*}, \tau\right)$$

$$\frac{i + i7^{*} \leq 64}{\vdash \text{setup\_dma\_store}\left(\mathrm{y}_{i}, i7^{*}\right) : DSS \rightarrow \left(\mathrm{y}_{i}, i7^{*}\right)}$$

$$\frac{\Gamma \vdash rr : at_1 \qquad fst(at_1) \equiv \mathrm{ptr}(p) \qquad \Omega \vdash p : \tau \qquad \Pi \vdash \left(Y, i7^{*}\right) : \tau_1 \qquad \tau_1 <: \tau}{\left(Y, i7^{*}\right); \Omega \vdash \text{start\_dma\_store}(rr) : \varepsilon \rightarrow \text{type\_preserving}}$$

$$\frac{\Gamma \vdash rr : at_1 \qquad fst(at_1) \equiv \mathrm{ptr}(p) \qquad \Pi \vdash \left(Y, i7^{*}\right) : \tau \qquad p \notin dom(\Omega) \qquad \vdash \Omega[p \mapsto \tau]}{\left(Y, i7^{*}\right); \Omega \vdash \text{start\_dma\_store}(rr) : \varepsilon \rightarrow (p, \tau)}$$

$$\vdash \text{wait\_dma\_store} : \text{type\_preserving} \rightarrow \varepsilon$$

$$\vdash \text{wait\_dma\_store} : (p, \tau) \rightarrow \varepsilon \mid \Omega \rightarrow \Omega[p \mapsto \tau]$$

conditional instructions:

$$\frac{\begin{array}{c}\vdash \iota : \Gamma_1 \rightarrow \Gamma_2 \mid u_1 \rightarrow u_2 \mid \Psi_1 \rightarrow \Psi_2 \mid \Sigma_1^{tmu} \rightarrow \Sigma_2^{tmu} \mid C_1 \rightarrow C_2 \mid wr_1 \rightarrow wr_2 \\ [rr \mapsto at_1] \in \Gamma_1 \qquad \Gamma_2 \equiv \Gamma_{21}[rr \mapsto at_2] \\ at_1 <: at_2\end{array}}{\vdash (\iota, c) : \Gamma_1 \rightarrow \Gamma_2 \mid u_1 \rightarrow u_2 \mid \Psi_1 \rightarrow \Psi_2 \mid \Sigma_1^{tmu} \rightarrow \Sigma_2^{tmu} \mid C_1 \rightarrow C_2 \mid wr_1 \rightarrow wr_2}$$

$$\frac{\vdash \iota : \Theta_1 \rightarrow \Theta_2}{\vdash (\iota, \varepsilon) : \Theta_1 \rightarrow \Theta_2}$$

signals:

$$\vdash \text{load}\langle tmu \rangle : \Sigma^{tmu} :: at \rightarrow \Sigma^{tmu} \mid \Gamma \rightarrow \Gamma[\mathrm{r4} \mapsto at]$$

conditional instructions with signals:

$$\frac{\vdash ci : \Theta_1 \to \Theta_2}{\vdash (ci, \varepsilon) : \Theta_1 \to \Theta_2}$$

$$\frac{\begin{array}{c}\vdash ci : \Gamma \to \Gamma[rr_1 \mapsto at_1] \mid u_1 \to u_2 \mid \Psi_1 \to \Psi_2 \mid C_1 \to C_2 \mid \Omega_1 \to \Omega_2 \mid wr_1 \to wr_2 \\ \vdash s : \Gamma \to \Gamma[\mathrm{r4} \mapsto at_2] \mid \Sigma_1^{tmu} \to \Sigma_2^{tmu}\end{array}}{\begin{array}{c}\vdash (ci, s) : \Gamma \to \Gamma[rr_1 \mapsto at_1][\mathrm{r4} \mapsto at_2] \mid u_1 \to u_2 \mid \Psi_1 \to \Psi_2 \\ \mid \Sigma_1^{tmu} \to \Sigma_2^{tmu} \mid C_1 \to C_2 \mid \Omega_1 \to \Omega_2 \mid wr_1 \to wr_2\end{array}}$$

labels:

$$\Phi \vdash l : \Phi(l)$$

instruction sequences:

$$\frac{\begin{array}{c}\vdash csi : \Theta_1 \to \Theta_2 \\ \Phi \vdash I : \Theta_2\end{array}}{\Phi \vdash csi \ ; I : \Theta_1}$$

$$\frac{\begin{array}{c}\vdash csi_1 : \Theta_1 \to \Theta_2 \\ \vdash csi_2 : \Theta_2 \to \Theta_3 \\ \vdash csi_3 : \Theta_3 \to \Theta_4 \\ \Phi \vdash l : \mathrm{code}(\Theta_5) \\ \Theta_4 <: \Theta_5\end{array}}{\Phi \vdash \mathrm{jmp}(l) \ ; csi_1 \ ; csi_2 \ ; csi_3 : \Theta_1}$$

$$\frac{\begin{array}{c} \vdash ci : \Theta_{11} \to \Theta_2 \qquad \vdash csi_1 : \Theta_2 \to \Theta_3 \\ \vdash csi_2 : \Theta_3 \to \Theta_4 \\ regctx(\Theta_2) \equiv regctx(\Theta_{11})[rwr \mapsto at] \\ rwr \not\equiv \mathrm{ra}n \\ rwr \not\equiv \mathrm{rb}n \qquad \Theta_1 \equiv \left(\Gamma_1, \Psi_1, u_1, \Sigma_1^{tmu}, C_1, \Omega_1, wr_1\right) \\ \Theta_{11} \equiv \left(\Gamma_1 \setminus \{\mathrm{ra14}, \mathrm{rb14}\}, \Psi_1, u_1, \Sigma_1^{tmu}, C_1, \Omega_1, wr_1\right) \\ \{\mathrm{ra14}, \mathrm{rb14}\} \notin regctx(\Theta_2) \\ \{\mathrm{ra14}, \mathrm{rb14}\} \notin regctx(\Theta_3) \\ \{\mathrm{ra14}, \mathrm{rb14}\} \notin regctx(\Theta_4) \end{array}}{\vdash (ci, \mathrm{thread\_end}) \ ; \ csi_1 \ ; \ csi_2 : \Theta_1}$$

$$\frac{\begin{array}{c} \vdash csi_1 : \Theta_1 \to \Theta_2 \qquad \vdash csi_2 : \Theta_2 \to \Theta_3 \\ \vdash csi_3 : \Theta_3 \to \Theta_4 \\ \Phi \vdash l : \mathrm{code}(\Theta_4) \qquad \Phi \vdash I : \Theta_5 \\ \Theta_4 <: \Theta_5 \end{array}}{\Phi \vdash \mathrm{if}\ qc\ \mathrm{jmp}(l) \ ; \ csi_1 \ ; \ csi_2 \ ; \ csi_3 \ ; \ I : \Theta_1}$$

programs:

$$\frac{\forall l \in dom(P) \,.\, \Phi \vdash P(l) : \Theta_l}{\Phi \vdash P}$$

It is defined that when the elements of at all have the same basic type b, it is convertible with $\mathrm{vec}(b, 16)$.

2.4. Auxiliary functions

Note that all free meta-variables are assumed to be fresh.

$$notfull\left(\Sigma^{tmu}\right) \stackrel{\text{def}}{=} \left(\Sigma^{tmu} \not\equiv at_1 :: at_2 :: at_3 :: at_4\right)$$

$$unwrap_{\Xi}\left(\mathrm{ptr}(p)\right) \stackrel{\text{def}}{=} \Xi(p)$$

$$fst\left([b_0, b_1, \ldots, b_{15}]\right) \stackrel{\text{def}}{=} b_0$$

$$map\left(f, [b_0, b_1, \ldots, b_{15}]\right) \stackrel{\text{def}}{=} [f(b_0), f(b_1), \ldots, f(b_{15})]$$

When an array type has the same 16 basic type, written $array(b)$:

$$array(b) \stackrel{\text{def}}{=} [b, b, ..., b]$$

$$inc\left(\mathrm{y}_{63}\right) \stackrel{\text{def}}{=} \mathrm{y}_0$$

$$inc\left(\mathrm{y}_n\right) \stackrel{\text{def}}{=} \mathrm{y}_{n+1} \text{ if } 0 \leq n \leq 62$$

$$regctx\big((\Gamma, \Psi, u, \Sigma, C, \Omega, wr)\big) \stackrel{\text{def}}{=} \Gamma$$

$size\_of(\tau)$ represents the size of a value of τ in bytes.

$$size\_of(b) \stackrel{\text{def}}{=} 4$$

$$size\_of(at) \stackrel{\text{def}}{=} 16 \times 4$$

$$size\_of(b \circ \Psi) \stackrel{\text{def}}{=} 4 + size\_of(\Psi)$$

$$size\_of(\varepsilon) \stackrel{\text{def}}{=} 0$$

$$size\_of\big(\mathrm{vec}(b, n)\big) \stackrel{\text{def}}{=} size\_of(b) \times n$$

$$dma\_load(Y, 1, \tau) \stackrel{\text{def}}{=} \big[Y \mapsto truncate(\tau)\big]$$

$$dma\_load\left(Y, i4^*, at\right) \stackrel{\text{def}}{=} [Y \mapsto at] \text{ if } i4^* \geq 2$$

$$truncate(at) \stackrel{\text{def}}{=} at$$

$$concat\big(\mathrm{vec}(b, m), \mathrm{vec}(b, n)\big) \stackrel{\text{def}}{=} \mathrm{vec}(b, m+n)$$

3. Future

- Any properties are not proved.
- There are many implicitness.
- The current definition is so conservative that it cannot serve practical use.
- The current definition may be incorrect or inconsistent.