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.

$\nu ::= n \mid rr$	operands
$n ::= i \mid f$	numbers
$r ::= rr \mid wr$	registers
$rr ::= rwr \mid \mathrm{uniform} \mid \mathrm{element\_number} \mid \mathrm{vpmr}$	readable registers
$urr ::= a \mid \mathrm{element\_number}$	unconstrained readable registers
$wr ::= rwr \mid \mathrm{uniforms\_address} \mid tmu \mid \mathrm{broadcast} \mid \mathrm{vpmw}$	writable registers
$rwr ::= ga \mid rf$	both readable and writable registers
$a ::= ga \mid sa$	accumulators
$ga ::= \mathrm{r0} \mid \mathrm{r1} \mid \mathrm{r2} \mid \mathrm{r3}$	general-purpose accumulators
$sa ::= \mathrm{r4} \mid \mathrm{r5}$	special-purpose accumulators
$rf ::= A \mid B$	register files
$A ::= \mathrm{ra0} \mid \ldots \mid \mathrm{ra31}$	locations in register file A
$B ::= \mathrm{rb0} \mid \ldots \mid \mathrm{rb31}$	locations in register file B
$tmu ::= \mathrm{tmu0} \mid \mathrm{tmu1}$	TMU
$vpmq ::= \varepsilon \mid (Y, i) :: vpmq$	VPM read queues
$Y ::= \mathrm{y}_0 \mid \ldots \mid \mathrm{y}_{63}$	VPM Y

$$\iota ::= \mathrm{rotate}(wr,\ rr,\ i4) \mid \mathrm{mov}(wr,\ \nu) \mid \mathrm{setup\_vpm\_read}\left(Y,\ i4^{*}\right) \mid \mathrm{setup\_vpm\_write}(Y) \mid \mathrm{setup\_dma\_load}\left(Y,\ i4^{*}\right) \mid \mathrm{start\_dma\_load}(rr) \mid \mathrm{wait\_dma\_load} \mid \mathrm{setup\_dma\_store}\left(Y,\ i7^{*}\right) \mid \mathrm{start\_dma\_store}(rr) \mid \mathrm{wait\_dma\_store}$$ instructions

$$cc ::= \mathrm{Z} \mid \mathrm{N} \mid \mathrm{C}$$ condition classes

$$c ::= \mathrm{set}(cc) \mid \mathrm{clear}(cc)$$ conditions

$$qc ::= \mathrm{all}(c) \mid \mathrm{any}(c)$$ quantified conditions

$$s ::= \varepsilon \mid \mathrm{load}\langle tmu \rangle$$ signals

$$ci ::= (\iota,\ \varepsilon) \mid (\iota,\ c)$$ conditional instructions

$$csi ::= (ci,\ s)$$ conditional instructions with signals

$$I ::= \mathrm{jmp}(l)\ ;\ csi\ ;\ csi\ ;\ csi \mid (ci,\ \mathrm{thread\_end})\ ;\ csi\ ;\ csi \mid csi\ ;\ I \mid \mathrm{if}\ qc\ \mathrm{jmp}(l)\ ;\ csi\ ;\ csi\ ;\ csi\ ;\ I$$ instruction sequences

$$R ::= \varepsilon \mid R[rr \mapsto arr]$$ register contexts

$$arr ::= [n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n]$$ arrays

$$U ::= \varepsilon \mid n \circ U$$ uniforms

$$T ::= \varepsilon \mid arr \mid arr :: arr \mid arr :: arr :: arr \mid arr :: arr :: arr :: arr$$ TMU

$$V ::= \varepsilon \mid V[Y \mapsto at]$$ VPM states

$$P ::= \varepsilon \mid P[l \mapsto I]$$ programs

$$M ::= (R,\ U,\ T,\ V,\ vpmq,\ P,\ I)$$ machines

2. Type system

The type syntax is defined as follows:

$$\tau ::= b \mid at \mid vt \mid \Psi$$ 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
$\Gamma ::= \varepsilon \mid \Gamma[rr \mapsto at]$	register context types
$\Psi ::= \varepsilon \mid b \circ \Psi$	uniforms types
$\Sigma ::= \varepsilon \mid at \mid at :: at \mid at :: at :: at \mid at :: at :: at :: at$	TMU types
$\Pi ::= \varepsilon \mid \Pi[Y \mapsto at]$	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
$\Theta ::= (\Gamma, \Psi, u, \Sigma, C, \Omega, wr)$	state types
$\Phi ::= \varepsilon \mid \Phi[l \mapsto \Theta]$	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 e-quality rules for p are not defined here. There are the abuses of notations of a form $x[y \mapsto z]$.

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

$$\frac{p = p_1}{\Xi\,[p_1 \mapsto \tau](p) \to_\Xi \tau}$$

$$\frac{p \neq p_1}{\Xi\,[p_1 \mapsto \tau](p) \to_\Xi \Xi(p)}$$

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:

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

subtype relations:

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

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

$$\Gamma <: \Gamma \qquad \frac{\Gamma_1 <: \Gamma_2}{\Gamma_1[rr \mapsto at] <: \Gamma_2}$$

pointers:

$$\frac{p \in dom(\Xi)}{\Xi \vdash p : \Xi(p)}$$

well-formed memory subsets:

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

registers:

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

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

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

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

VPM:

$$\frac{Y \in dom(\Pi)}{\Pi \vdash Y : \Pi(Y)}$$

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

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

instructions:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

conditional instructions:

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

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

signals:

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

conditional instructions with signals:

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

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

labels:

$$\frac{l \in dom(\Phi)}{\Phi \vdash l : \Phi(l)}$$

instruction sequences:

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

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

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

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

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.

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

$$unwrap_{\Xi}(\mathrm{ptr}(p)) \stackrel{\text{def}}{=} \Xi(p) \text{ if } p \in dom(\Xi)$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.