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

$$\begin{aligned}\iota ::= {} & \mathrm{rotate}(wr,\ rr,\ i4) \mid \mathrm{mov}(wr,\ \nu) \mid \mathrm{setup\_vpm\_read}\big(Y,\ i4^{*}\big) \\ & \mid \mathrm{setup\_vpm\_write}(Y) \mid \mathrm{setup\_dma\_load}\big(Y,\ i4^{*}\big) \mid \mathrm{start\_dma\_load}(rr) \\ & \mid \mathrm{wait\_dma\_load} \mid \mathrm{setup\_dma\_store}\big(Y,\ i7^{*}\big) \mid \mathrm{start\_dma\_store}(rr) \\ & \mid \mathrm{wait\_dma\_store} \qquad \text{instructions}\end{aligned}$$

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

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

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

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

$$ci ::= (\iota,\ \varepsilon) \mid (\iota,\ c) \qquad \text{conditional instructions}$$

$$csi ::= (ci,\ s) \qquad \text{conditional instructions with signals}$$

$$\begin{aligned}I ::= {} & \mathrm{jmp}(l)\ ;\ csi\ ;\ csi\ ;\ csi \mid (ci,\ \mathrm{thread\_end})\ ;\ csi\ ;\ csi \mid csi\ ;\ I \\ & \mid \mathrm{if}\ qc\ \mathrm{jmp}(l)\ ;\ csi\ ;\ csi\ ;\ csi\ ;\ I \qquad \text{instruction sequences}\end{aligned}$$

$$R ::= \varepsilon \mid R[rr \mapsto arr] \qquad \text{register contexts}$$

$$arr ::= [n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n,\ n] \qquad \text{arrays}$$

$$U ::= \varepsilon \mid n \circ U \qquad \text{uniforms}$$

$$T ::= \varepsilon \mid arr \mid arr :: arr \mid arr :: arr :: arr \mid arr :: arr :: arr :: arr \qquad \text{TMU}$$

$$V ::= \varepsilon \mid V\big[Y \mapsto at\big] \qquad \text{VPM states}$$

$$P ::= \varepsilon \mid P\big[l \mapsto I\big] \qquad \text{programs}$$

$$M ::= \big(R,\ U,\ T,\ V,\ vpmq,\ P,\ I\big) \qquad \text{machines}$$

2. Type system

The type syntax is defined as follows:

$$\tau ::= b \mid at \mid vt \mid \Psi \qquad \text{types}$$

$p ::= nat \mid \alpha \mid p + p \mid nat \times p$ 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 equality rules for p are not defined here. There are the abuses of notations of a form $x[y \mapsto z]$.

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

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

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

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:

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

subtype relations:

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

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

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

pointers:

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

well-formed memory subsets:

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

registers:

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

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

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

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

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:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{i + i4^* \le 64}{\vdash \text{setup\_dma\_load}\big(\text{y}_i,\ i4^*\big) : \mathit{DLS} \to \big(\text{y}_i,\ i4^*\big)}$$

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

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

$$\frac{i + i7^* \le 64}{\vdash \text{setup\_dma\_store}\big(\text{y}_i,\ i7^*\big) : \mathit{DSS} \to \big(\text{y}_i,\ i7^*\big)}$$

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

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

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

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

conditional instructions:

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

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

signals:

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

conditional instructions with signals:

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

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

labels:

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

instruction sequences:

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

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

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

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

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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.