1 Overview

1. FP-Units:
    - FX-arithmetic
    - theory of rounding
    - + -
    - * /
    - rounding
2. Automotive systems:
    - ECU
    - Bus
    - r.t. OS
    - WCET

2 FP-Units

Definition 2.1 (binary numbers)

- *bit vectors:*
$$\{0,1\}^n = a[n-1:0] = a_{n-1}...a_0$$
- *value interpeted as binary number*
$$\langle a \rangle = \sum_{i=0}^{n-1} a_i * 2^i \ \in [0:2^n-1]$$

components of circuits

- gates: $\wedge, \vee, \oplus, \neg$
- registers: input: n-bit, output: n-bit, Rce: clock enable

Figure 1: Gates

Definition 2.2 (n-bit adder)
$\langle c_{out}, s \rangle = \langle a \rangle + \langle b \rangle + c_{in}$

Figure 2: n-bit Adder

Definition 2.3 (full adder (FA))
1 bit adder
$s = a \oplus b \oplus c$
$c' = (a \wedge b) \vee (a \wedge c) \vee (b \wedge c)$

Figure 3: Full Adder

Figure 4: n-bit Carry Chain Adder

2.1 Carry Chain Adder

construction with n full adder; recursive definition

Lemma 2.1 *This is a n-bit adder*

Auxiliary result: $\langle a[n-1:0]\rangle = \langle a[n-1:h]\rangle * 2^h + \langle a[h-1:0]\rangle$
$Proof$: Ind(n)

$n = 1$: okay
$n-1 \Rightarrow n$:
$\langle a\rangle + \langle b\rangle + c_0$
$= a_{n-1} * 2^{n-1} + \langle a[n-2:0]\rangle + b_{n-1} * 2^{n-1} + \langle b[n-2:0]\rangle + c_0$
$= a_{n-1} * 2^{n-1} + b_{n-1} * 2^{n-1} + \langle c_{n-1}, s[n-2:0]\rangle$
$= (a_{n-1} + b_{n-1} + c_{n-1}) * 2^{n-1} + \langle s[n-2:0]\rangle$
$= \langle c_n s_{n-1} s[n-2:0]\rangle = \langle c_n s[n-1:0]\rangle$

Definition 2.4 (Delay and Cost) *Let S be a circuit, then we define:*

- **Delay**
  $D(p) = \sum_g D(g)$ *where p is a path and g gate on path*
  $D(\wedge), D(\vee), ...technology$
  $D(S) = max\{D(p) | p\ path\ in\ S\}$

- **Cost**
  $C(S) = \sum_g C(g)$ *where g gate in circuit S*

Cost and Delay CCA
$C(CCA_n) = n * C(FA)$ $\Rightarrow$ cheap
$D(CCA_n) \leq n * D(FA)$ $\Rightarrow$ very slow

Auxiliary circuit n-multiplexer (n-mux)

$$z := \begin{cases} x & : s = 0 \\ y & : s = 1 \end{cases}$$

Figure 5: 1-bit multiplexer

Figure 6: n-bit multiplexer

2.2 Conditional Sum Adder $\mathbf{COS_n}$

$COS_n = FA$
$n = 2^k$
$a_H = a[n-1 : \frac{n}{2}]$ high bits
$b_L = a[\frac{n}{2} - 1 : 0]$ low bits

Figure 7: n-bit conditional sum adder

Delay

$$\begin{aligned} D(1) &= D(FA) \\ D(n) = D(COS_n) &= D(\tfrac{n}{2}) + D(Mux) \\ &= 2 * D(Mux) + D(\tfrac{n}{4}) \\ &\vdots \\ &= i * D(Mux) + D(\tfrac{n}{2^i}) = k * D(Mux) + D(1) \end{aligned}$$

Lemma 2.2 (Delay $\mathbf{COS_n}$)

$$D(n) = \log n * D(Mux) + D(FA)$$

Cost ignoring muxes

$$\begin{aligned} C(1) &= C(FA) \\ C(n) &= 3 * C(\tfrac{n}{2}) = 3^{\log n} * C(1) \\ 3^{\log n} &= (2^{\log_2 3})^{\log_2 n} = 2^{(\log_2 3) * \log_2 n} \\ &= (2^{\log_2 n})^{\log_2 3} = n^{1.7...} \end{aligned}$$

2.3 Compound Adder

Definition 2.5 (n-bit Compound Adder $\mathbf{CPA_n}$)

$$\langle s \rangle = \langle a \rangle + \langle b \rangle$$

$$\langle s' \rangle = \langle a \rangle + \langle b \rangle + 1$$

Figure 8: n-bit Compound Adder

Cost
$C(n) = C(CPA_n) = 2 * C(\frac{n}{2}) + \underbrace{2 * C((\frac{n}{2} - 1) - Mux)}_{(n+2)*C(Mux)}$
$C(1) \leq 2 * C(FA)$
$C(n) \leq Q * n \log n$ for some Q

2.4 Carry Lookahead Adder CLA

$D(n) = O(\log n)$
$C(n) = O(n)$

Definition 2.6 (parallel prefix computation)
$\circ : M \times M \rightarrow M$ *associative*
$PP_\circ(n) : M^n \rightarrow M^n$
$PP_\circ(n) : (X_{n-1}, \ldots, X_0) = (Y_{n-1}, \ldots, Y_0)$
$Y_i = X_i \circ \ldots \circ X_0$
$Y_0 = X_0$

Example: $\circ = +$
$Y_i = \sum_{j \leq i} X_i$

Definition 2.7 (PP-Circuit) *Let* $Y = PP_0(n)(X)$

Figure 9: $PP_o(n) - Circuit$

Figure 10: PP_1

Construction
$\mathbf{PP_1}$:$Y_0 = X_0$
$\mathbf{PP_n}$:

$$\begin{aligned} Y'_i = X'_i \circ \ldots \circ X'_0 &= (X_{2i+1} \circ X_{2i}) \circ \ldots \circ (X_1 \circ X_0) \text{ ind. hyp.} \\ &= \Big(((X_{2i+1} \circ X_{2i}) \circ \ldots) \circ X_0 \Big) \\ &= Y_{21+1} \end{aligned}$$

$Y_{2i} = X_{2i} \circ Y'_{2i-1}$

Figure 11: PP_n

Delay
$D(1) = 0$
$D(n) = D(\frac{n}{2}) + 2 \leq 2 * (\log_2(n) - 1)$

Cost
$C(1) = 0$
$C(n) \leq C(\frac{n}{2}) + n \leq 2 * n$

Let $a, b \in \{0,1\}^n$, $c_{in} \in \{0,1\}$ and i, j $i < j$:
$P_{i,j}(a,b) : \ \langle a[j:i] \rangle + \langle b[j:i] \rangle \ = \ \langle 1^{j-i+1} 1 \rangle$ **Propagate**
$G_{i,j}(a,b) : \ \langle a[j:i] \rangle + \langle b[j:i] \rangle \ \geq \ \langle 10^{j-i+1} \rangle$ **Generate**
$G_{0,j}(a,b,c_{in}) : \ \langle a[j:i] \rangle + \langle b[j:i] \rangle + c_{in} \ \geq \ \langle 10^{j-i+1} \rangle$
$c_i = g_{0,i-1}$
$s_i = a_i \oplus b_i \oplus c_i$

Lemma 2.3 *Let* $i \leq j \leq k$

$$P_{i,k}(a,b) = P_{i,j}(a,b) \wedge P_{j+1,k}(a,b)$$
$$G_{i,k}(a,b,c_{in}) = G_{j+1,k}(a,b,c_{in}) \vee G_{i,j}(a,b) \wedge P_{i+1,k}(a,b)$$

$(q,p) = (q_1 \vee q_0 \wedge p_1, p_0 \wedge p_1) = (q_1,p_1) \circ (q_0,p_0)$
$\circ : \{0,1\}^4 \rightarrow \{0,1\}^2$

Lemma 2.4 $\circ$ *is associative*

Figure 12: ∘-Circuit

Let $PP_0(n)\Big((P_{n-1,n-1}, G_{n-1,n-1}) \dots (P_{0,0}, G_{0,0})\Big)$
$= PP_0(n)\Big((P_{n-1}, G_{n-1}) \dots (P_0, G_0)\Big)$

$i > 0\ q_{i,i} = a_i \wedge b_i$
$i = 0\ q_{0,0} = (a_0 \wedge b_0) \vee (a_0 \wedge c_{in}) \vee (b_0 \wedge c_{in}) \Rightarrow G_i = q_{0,i} = c_{i+1}$
$\quad p_{i,i} = a_i \oplus b_i$

Carry-Lookahaed-Adder:
$D(CLA_n) = O(logn)$
$C(CLA_n) = O(n)$

O-Notation
$f, g \in \mathbb{N}_0 \to \mathbb{N}_0$
$f(n) \in O(g(n)) \Leftrightarrow \exists e > 0, n_0 : \forall n \geq n_0 : f(n) \leq c * g(n)$

2.5 Multipliers

Excursion: Substraction of binary numbers

Definition 2.8 (two's complement number)
Let $a \in \{0,1\}^n$
$[a] = -a_{n-1}2^{n-1} + \langle a[n-2:0]\rangle \in \{-2_{n-1} \dots 2^{n-1} - 1\}$

Lemma 2.5 *Let* $a \in \{0,1\}^n$*:*

- $[0a] = \langle a \rangle$
- $[a] = [a_{n-1}a]$
- $[a] \leq 0 \Leftrightarrow a_{n-1} = 1$
- $\langle a \rangle = [a] mod 2^n$
- $-[a] = [\bar{a}] + 1$

$Proof([a] = [a_{n-1}a])$:

$$\begin{aligned}[a_{n-1}a] &= -a_{n-1} * 2^n + \langle a \rangle \\ &= -a_{n-1} * 2^n + 2_{n-1} * 2^{n-1} + \langle a[n-2:0]\rangle \\ &= -a_{n-1} * 2^{n-1} + \langle a \rangle \\ &= [a]\end{aligned}$$

$Proof(\langle a\rangle = [a] mod 2^n)$:

$$\begin{aligned}\langle a\rangle - [a] &= a_{n-1} * 2^{n-1} - (-a_{n-1} * 2^{n-1}) \\ &= a_{n-1} * 2^n\end{aligned}$$

$Proof(-[a] = [\bar{a}] + 1)$:

$$\begin{aligned}[\bar{a}] &= -\bar{a_{n-1}} * 2^n + \textstyle\sum_{i=0}^{n-2} \bar{a}[i] * 2^i \\ &= -(1 - a_{n-1}) * 2^{n-1} + \textstyle\sum_{i=0}^{n-2}(1 - a[i]) * 2^i \\ &= a_{n-1} * 2^{n-1} - \langle a[n-2:0]\rangle - 2^{n-1} + \textstyle\sum_{i=0}^{n-1} 2^i \\ &= -[a] - 1\end{aligned}$$

Substraction

Let $x, y \in 0, 1^n$

$\langle x\rangle - \langle y\rangle = \langle x\rangle + \langle y\rangle + 1 mod 2^n$

$Proof$:

$$\begin{aligned}\langle a\rangle - \langle y\rangle &= \langle x\rangle - [0y] \\ &= \langle x\rangle + [1\bar{y}] + 1 \\ &= \langle x\rangle - 2^n + \langle \bar{y}\rangle + 1 \\ &= \langle x\rangle + \langle \bar{y}\rangle + 1 mod 2^n\end{aligned}$$

Figure 13: n-bit Substraction

Multiplication

Definition 2.9 ((n,m)-Multiplier)

Figure 14: n,m Multiplier

$$\langle a\rangle * \langle b\rangle \leq (2^n - 1) * (2^n - 1) < 2^{n+m} - 1$$

$\langle s\rangle = \langle a\rangle * \langle b\rangle$

School Method

$$\begin{aligned}\langle a\rangle * \langle b\rangle &= \langle a\rangle * \textstyle\sum_{i=0}^{m-1} b_i * 2^i \\ &= \textstyle\sum_{i=0}^{n-1} \langle a\rangle * b_i * 2^i \\ &= \textstyle\sum_{i=0}^{n-1} \langle (a \wedge b0^i)\rangle\end{aligned}$$

i-th partial products: $\langle a \rangle * b_i * 2^i$
$(a \wedge b) = (a_{n-1} \wedge b_i, \ldots a_0 \wedge b_i) \in \{a, 0^n\}$
m many $(n+m)$-bit adders $\rightarrow$ reduce depth to $O(log(n+m))$
bounding the wire of adders
Sums of k consecutive partial products starting with the i-th bis

$S_{i,k} = \sum_{t=j}^{j+k-1} \langle a \rangle * b_t * 2^t$
$\langle a \rangle * 2^j * \langle b[j+k-1:j] \rangle$
$< 2^{n+k+j}$

$s_{0,1} = \langle a \wedge b_0 \rangle$
$s_{i,1} = \langle (a \wedge b) 0^i \rangle$

Figure 15: Partial Product $S_{0,1}$ and $S_{i,1}$

Multiplier

Figure 16: Mutliplier

m many n-bit adder suffice
if $m = n$ $((n * m) - Mult) \leq n^2 * C(FA))$

Reducing depth
Use tree of adders:

Figure 17: tree of adders

Delay: $\lceil logm \rceil * O(log(m+n))$
if $m = n$: $O((logn)^2)$

$\langle s \rangle + \langle t \rangle = \langle a \rangle + \langle b \rangle + \langle c \rangle$

$Proof:$

Figure 18: n-CSA

Definition 2.10 (n-bit carry-save adder (n-CSA))

Figure 19: Construction

$$\begin{aligned}\langle a\rangle + \langle b\rangle + \langle c\rangle &= \sum_{i=0}^{n-1}(a_i + b_i + c_i) * 2^i \\ &= \langle s[n-1:0]\rangle + \sum_{i=0}^{n-1} t_{i+1} * 2^{i+1} \\ &= \langle s\rangle + \sum_{j=0}^{n} t_j + 2^j + t_0 * 2^0 \\ &= \langle s\rangle + \langle t\rangle\end{aligned}$$

3/2 Adder = CSA Adder

4/2 Adder

Figure 20: 4/2-Adder

$\langle s\rangle + \langle t\rangle = \langle a\rangle + \langle b\rangle + \langle c\rangle + \langle d\rangle mod 2^n$
$\#leaves = m/4$
$\#levels = log(m/2) = log(m) - 2$
$D(4/2Adder) \leq 2 * D(FA)$
$D(Mult) \leq 2 * (log(m-2)) * D(FA) + O(log(n+m))$

$S_{i,k} = \sum_{j=1}^{i+k-1} \langle a\rangle b_j * 2^j$
$\langle s\rangle + \langle t\rangle = \langle a\rangle + \langle b\rangle + \langle c\rangle$ mod 2^n
$\langle s'\rangle + \langle t'\rangle = \langle s\rangle + \langle t\rangle + \langle d\rangle$ mod 2^n

Floating Point: 53,55 M=64
$2^{\lceil log(m)\rceil} = M$

$3(m/4 - a) + 4a = m$ $a = m - 3m/4$
$\Rightarrow m = 55, a = 7$
Question: # FA in Tree + top layer
Theorem: $n^2 + O(mlog(m))$
4/2 Adders are composed of 3/2 Adders
x-3/2 Adders have length $x = n$ regulars FA's $+l$excess FA's

Figure 21: Tree of 4/2 Adder

Figure 22: partial product m+k$||0^i$

Theorem: # excess FA's $= O(m * log(m))$

- top level 3/2 Adders: no excess Adders
- top level 4/2 Adders: no excess Adders
- node v in T:

Every node in T computes a sum $S_{i,x}$, $x = w(v)$
v root: $w(v) = m$
top level: w(v) $\in \{3, 4\}$
l(v): left parent of v
r(v): right parent of v
# of excess FA's $= 2 * \sum_{\text{v in t}} w(l(v))$
$= 2 * \sum_{\text{level } \lambda} \sum_{\text{v in level } \lambda} w(l(v))$

Lemma 2.6
$\forall\lambda$: *weights of nodes in level* λ *increase from left to right.*
second λ *starts with* $\lambda = M + 1$

$$\Rightarrow w(l(v)) \leq w(r(v))$$

$$\Rightarrow 2 * \sum_{u \text{ in level}\lambda} w(l(v)) \leq \sum_{u \text{ in level } \lambda} w(l(u)) + w(r(u))$$

$$\Rightarrow \sum_{u \text{ in level}\lambda+1} w(v) = m$$

# of excess FA $\leq \mu * m < m * log(m)$

Booth Recoding
$\langle a \rangle * \langle b \rangle$
$\langle b \rangle = 2 * \langle b \rangle - \langle b \rangle = \langle b0 \rangle + \langle 0b \rangle$

Figure 23: 4/2-Adder tree

Figure 24: Excess Adders:

Figure 25: Booth digits

$\langle b\rangle = 2 * \langle b\rangle - \langle b\rangle = \sum_{j=0}^{m'-1}$ where $m' = \lceil (m+1)/2 \rceil$

$B_{2j} = 2b_{2j} + b_{2j-1} - 2 * b_{2j+1} - b_{2j} = -2b_{2j+1} + b_{2j} + b_{2j-1} \in \{-2, \ldots, 2\}$

$\langle a\rangle * \langle b\rangle = \sum \langle a\rangle * B_{2j} * 4^j$

$$C_{2j} = \langle a\rangle * B_{2j}$$
$$D_{2j} = |\langle a\rangle * B_{2j}| in \{0, \ldots, 2^{n+1} - 1\}$$
$$d_{2j} = bin_{n+1}(D_{2j}$$

where $\langle a\rangle = x$, $a \in \{0,1\}^n, a = bin_n(x)$

$$s_{2j} = \begin{cases} 0 & B_{2j} \geq 0 \\ 1 & B_{2j} < 0 \end{cases}$$

$$\langle a\rangle * \langle b\rangle = \sum_{j=0}^{m'-1} (-1)^{s_{2j}} D_{2j} * 4^j$$

$$0 < E_{2j} = C_{2j} + 3 * 2^{n+1}$$
$$0 < E_0 = C_0 + 4 * 2^(n+1)$$

Figure 26: Error Terms + D_{2j}'s

Error Terms: $\sum_{j>0}(3 * 2^{n+1}) * 4^j + 4 * 2^{n+1} = 0 \bmod 2^{n+m}$
$\langle a\rangle * \langle b\rangle = \sum (E_{2j} * 4^j \bmod 2^{n+m}$

$e_{2j} = bin_{n+3} E_{2j}$
$e_0 = bin_{n+3} E_0$

Lemma 2.7 $\langle e_{2j}\rangle = \langle 1\bar{s}_2 j, d_{2j} \oplus s_{2j}\rangle + s_{2j}$
$\langle e_0\rangle = \langle \bar{s_0} s_0 s_0, d_0 \oplus s_0\rangle + s_0$

$s_{2j} = 0$: nothing happens
$\langle 110^{n+1} \rangle - \langle 00 d_{2j} \rangle$
$= \langle 110^{n+1} \rangle + \langle 11 \bar{d_{2j}} \rangle + 1 \bmod 2^{n+3} = \langle 110 \bar{d_{2j}} \rangle + 1 = \langle 10 \bar{d_{2j}} \rangle + 1 \bmod 2^{n+3}$

Add sign-bits to the d_{2j}'s:

$$g_{2j} = e_{2j} 0 s_{2j-2}$$

$\langle a \rangle * \langle b \rangle = \sum \langle g_{2j} \rangle * 4^j \bmod 2^{n+m}$

(n,m)-multiplier
cost: $n^2 * (C(FA) + C(1))$

(n,m)-Booth multiplier
cost: $n^2(C(FA)/2 + C(BSL)/2) + O(n^2)$
$bx_{2j} = 1 \Leftrightarrow \|B_{2j} = x\|\ xin\{1, 2\}$
computed by Booth-Recoder m' many needed
Booth selection logic(Fig: 2.38): $\leq m' * (n + 5) = n^2/2 + o(n)$
Example Technology MP00

	$\neg$	$\wedge, \vee, \neg\wedge, \neg\vee$	$\oplus$	1-mux
Cost	1	2	4	3
Delay	1	2	2	2

Figure 27: Full-Adder

$C(FA) = 14$
$C(BSL) = 10 \Rightarrow n^2 * (14 + 2) \geq n^2 * (7 + 5) + o(n^2)$
today:
array:
no Booth: $(n - 1) * D(FA) + O(log(n))$
Booth: $^n/2 * D(FA) + O(log(n))$
trees:
saving only $O(1)$ in gate model

Figure 28: Multiplication Tree: area $O(n^2 log(n))$

2.6 Floating Point Units

IEEE-standard; theory of rounding (2 Weeks)
FPU construction (4 Weeks)

Definition 2.11 (binary fraction)

$$\langle a[n-1:0].b[1:p]\rangle = \sum_{i=0}^{n-1} a_i * 2^i + \sum_{j=1}^{p} b_j * 2^{-j} = \langle ab\rangle * 2^{-p}$$

Definition 2.12 (two's complement fraction)

$$[a[n-1:0].b[1:p]] = -a_{n-1} * 2^{n-1} + \langle a[n-2:o].b[1:p]\rangle = [ab] * 2^{-p}$$

Definition 2.13 (biased integer representation)
$e[n-1:0] \in \{0,1\}^n$
$\{0^n, 1^n\}$
$[|e|]_{bias} = \langle e\rangle - bias_n \in \{e_{min}, \ldots, e_{max}\}$
$bias_n = 2^{n-1} - 1$
$e_{min} = 1 - (2^{n-1} - 1) = 2 - 2^{n-1} = -2^{n-1} + 2$
$e_{max} = 2^n - 2 - (2^{n-1} - 1) = 2^{n-1} - 1$

$\{e_{min}, \ldots, e_{max}\} \subseteq T_n$
biased integer $x[n-1:0] \overset{convert}{\rightarrow}$ 2's complement $y[n-1:0]$
Solve for y:

$$[|x|]_{bias} = [y] \Leftrightarrow \langle x\rangle - (2^{n-1} - 1) = -y_{n-1} * 2^{n-1} + \langle y[n-2:0]\rangle$$

$$\langle x\rangle + 1 = 2^{n-1} * (1 - y_{n-1}) + \langle y[n-2:0]\rangle = \langle \bar{y}_{n-1} y[n-2:0]\rangle$$

Solve for x:($y \in \{e_{min}, \ldots, e_{max}\}$

$$\langle x\rangle = [y] + 2^{n-1} - 1 = [y] + \langle 01^{n-1}\rangle = \langle y\rangle + \langle 01^{n-1}\rangle mod 2^n$$

Definition 2.14 (IEEE-FP-Number)
$(s, e[n-1], f'[1:p-1])$
$s \in \{0,1\}$ *sign bit*
e: exponent
f': almost significant fraction

$$(n,p) = \begin{cases} (6,24) & \text{single precision} \\ (11,53) & \text{double precision} \end{cases}$$

$$[\![s,e,f']\!] = \begin{cases} (-1)^s * 2^{[\![e]\!]_{bias}} * \langle 1.f'\rangle & : e \in \{0^n, 1^n\} \\ (-1)^s * 2^{e_{min}} * \langle 0.f'\rangle & e = 0^n \\ (-1)^s * \infty & : e = 1^n \wedge f' = 0^{p-1} \\ NaN & : e = 1^n \wedge f' \neq 0^{p-1} \end{cases}$$

$$f[0] = \begin{cases} 1 & e \notin \{0^n, 1^n\} \\ 2 & e = 0^n \end{cases}$$

$$\text{exponent } 2^{e_{min}} \begin{cases} e = 0^{n-1} \\ e = 0^n \end{cases}$$

f:normal if $f[0] = 1$
f:denormal if $f[0] = 0$

$$(s, e, f) \begin{cases} \text{normal} e \in \{0^n, 1^n\} \\ \text{denormal} e = 0^n \end{cases}$$

$f[0] \leq \langle f \rangle \leq f[0] + \langle f[1:p-1] \rangle 2^{-1(p-1)} \leq f[0] + 1 - 2^{-(p-1)}$ $\Re$: representable

Figure 29: representable floating point numbers

finite numbers
$\Re_\infty = \Re \cup \{+\infty, -\infty\}$
$r : \Re \to \Re_\infty$ rounding functions
$r_v(x) = min\{y \in \Re_\infty \mid x \leq y\}$ round up
$r_v(x) = min\{y \in \Re_\infty \mid x \geq y\}$ round down

$$r_z(x) = \begin{cases} r_d(x) & : x \geq 0 \\ r_u(x) & : x < 0 \end{cases}$$

$let y = [\![s, e, f']\!]$
y is even if $f'[p-1] = 0$
$r_u(x) = \infty \leftrightarrow x > X_{max}$
$r_d(x) = \infty \leftrightarrow x < -X_{max}$

$$r_{ne} = \begin{cases} X_{max} & : X_{max} \leq x < X^x_{max} \\ +\infty & : x \geq X^*_{max} \\ -\infty & : x \leq X^*_{max} \\ -X_{max} & : X^*_{max} < x \leq X_{max} \end{cases}$$

$\circ : \mathbb{R}^2 \to \mathbb{R}$ $\circ_I : \mathbb{R}^2 \to \Re_\infty$

$$(x \circ_I y) = r(x \circ y)$$

complement $x \to x'$ $y \to y'$

$$r(x \circ y) = r(x' \circ y')$$

3 Theory of Rounding

Goal: abstract from the binary representation as long as possible

Definition 3.1 (Factoring (s,e,f))

$s \in 0, 1$ $\quad sign\ bit$
$e \in \mathbb{Z}$ $\quad exponent$
$f \in \mathbb{R}^{\geq 0}$ $\quad significant$

f **normal** if $f \in [1, 2)$
f **denormal** if $f \in [0, 1)$
(s, e, f) normal if f normal
(s, e, f) denormal if f denormal

$$[\![s, e, f]\!] = (-1)^s * 2^e * f$$

special factorings:

$$(s, \infty, 0) : [\![s, \infty, 0]\!] = (-1)^s * \infty$$

Definition 3.2 (IEEE normal)
(s, e, f) *is IEEE normal if*

$$\begin{cases} [\![s, e, f]\!] \geq 2^{e_{min}} \\ \Rightarrow (s, e, f)\ is\ normal \end{cases}$$

or

$$\begin{cases} [\![s, e, f]\!] < 2^{e_{min}} \\ \Rightarrow (s, e, f)\ is\ denormal \end{cases}$$

3.1 Normalization Shifts

Definition 3.3 ($\hat{\eta}(\mathbf{x})$)

$$\hat{\eta}(x) = \begin{cases} (sign(x), \infty, 0) & : x \in \{\pm\infty\} \\ (sign(x), e = \lfloor \log[2](x) \rfloor, |x| * 2^{-e}) & : otherwise \end{cases}$$

$\hat{\eta}$ yields a normal factoring for $x \neq 0$

Definition 3.4 ($\eta(\mathbf{x})$) *x with $\hat{\eta}(x) = (\hat{s}, \hat{e}, \hat{f})$* $x \neq 0$

$$\eta(x) = \begin{cases} (\hat{s}, \hat{e}, \hat{f}) & : |x| \geq 2^{e_{min}} \\ (\hat{s}, e_{min}, \hat{f} * 2^{\hat{e} - e_{min}}) & : |x| < 2^{e_{min}} \\ (0, e_{min}, 0) & : x = 0 \end{cases}$$

$\eta(x)$ yields an IEEE-normal factoring for any $x \in \mathbb{R} \backslash \{\pm\infty\}$

Properties:

- $[\![\hat{\eta}(x)]\!] = x\ for\ x \neq 0$
- $[\![\eta(x)]\!] = x$
- $\hat{\eta}(x) = \eta(x)\ for\ |x| \geq 2^{e_{min}}$

3.2 α-equivalence, α-representation, sticky-bit computation

Goal: recipe to compute r(x)

Let $q \in \mathbb{Z}$

Figure 30: α interval

$$\left.\begin{array}{l}(q*2^{-\alpha},\ldots,(q+1)*2^{-\alpha}) \\ \{q*2^{-\alpha}\}\end{array}\right\} \text{for all } q \in \mathbb{Z} \text{ yields partitions of } \mathbb{R}$$

Definition 3.5 (α-equivalence) $x, y \in \mathbb{R}$ *α-equivalent if*

$$x = y$$

or

$$\exists q : x \in (q*2^{-\alpha},\ldots,(q+1)*2^{-\alpha}) \; and \; y \in (q*2^{-\alpha},\ldots,(q+1)*2^{-\alpha})$$

Notation: $x =_\alpha y \Leftrightarrow$ x and y are α-equivalent.
Clearly: $q = q_\alpha(x) = \lfloor x * 2^{-\alpha} \rfloor$

Definition 3.6 (α-representative) $[x]_\alpha$ *α-representative of x*

$$[x]_\alpha = \begin{cases} x & : x = q_\alpha(x) * 2^{alpha} \\ (q_\alpha(x) + \frac{1}{2}) * 2^{-alpha} & : otherwise \end{cases}$$

Note: each $[x]_\alpha$ can be written as a binary fraction $\langle a[n-1:0].b[1:\alpha+1] \rangle = [x]_\alpha$

Properties:

- $x =_{[\alpha]} y \Leftrightarrow [x]_\alpha = [y]_\alpha$ (representative equivalence)
- $x =_{[\alpha]} y \Leftrightarrow -x =_\alpha -y, [-x]_\alpha = [-y]_\alpha$ (negation)
- $x =_{[\alpha]} y \Leftrightarrow 2^e * x =_{\alpha-e} 2^e * y, [2^e * x]_\alpha = 2^e * [x]_{\alpha-e}$ (skaling)
- $x =_{[\alpha]} y \Leftrightarrow x + k * 2^{-\alpha} = y + k * 2^{-\alpha} \; for \; k \in \mathbb{Z}$ (translation)
- $x =_{[\alpha]} y \Leftrightarrow x =_{\alpha-k} y \; k \geq 0$ (coarsening)

Lemma 3.1 (rounding with limited exponent range)
$x \in \mathbb{R}, \eta(x) = (s, e, f), r$*: an IEEE rounding function*

1. $r(x) = r([x]_{p-e})$

2. $\eta([x]_{p-e}) = (s, e, [f]_p)$

3. $x' =_{p-e} x \Rightarrow r(x) = r(x')$

Proof:
Part 1:

$$|x| \in \begin{cases} [2^e, 2^{e+1}) & : fnormal \\ (0, 2^e) & : fdenormal \end{cases}$$

In this interval, all representatives have a distance $d = 2^{e-(p-1)}$
$\Rightarrow \exists q : \ |x| \in [y, z) \ for y = q * 2^{e-(p-1)}, z = (q+1) * 2^{e-(p-1)}$
Obviously, $r(x) \in \{y, z\}$
To decide whether $r(x) = y$ or $r(x) = z$ it suffices to know $[x]_{p-e}$ instead of x

Figure 31: y-z-Interval

Part 2:

$[x]_{p-e} = [(-1)^s * 2^e * f]_{p-e} = (-1)^s * [2^e * f]_{p-e} = (-1)^s * 2^e * [f]_p$
$(s, e, [f[_p$ is a factoring of x
It is IEEE normal since

- (s, e, f) is normal
- $|x| \geq 2^{e_{min}} \Leftrightarrow |[x]_{p-e}| \geq 2^{e_{min}}$
- f is normal $\Leftrightarrow [f]_p$ is normal

Part 3:

$r(x) = r([x]_{p-e})$
$r(x') = r([x]_{p-e})$
$x' =_{p-e} x \Leftarrow [x]_{p-e} = [x']_{p-e}$

Computing a p-representative from a binary fraction is easy (sticky bit computation)

Lemma 3.2 (sticky bit computation) $f = \underbrace{a[n-1:0].f[1:p]}_{g} \underbrace{f[p+1:v]}_{OR-tree \Rightarrow s}$

$$[f]_p = \langle g, s \rangle$$

$g = a[n-1:0].f[1:p]$
$s = \bigvee_{i=p+1}^{v} f_i$ *sticky bit*

Proof: Exercise

Rounding with unlimited exponent

$$R = 0 \cup \{(-1)^s * 2^e * f | f \in [1, 2)\}$$

analogous to r: $\mathbb{R} \to \mathbb{R}_\infty$
define:

$$\hat{r} : \mathbb{R} \to \hat{\mathbb{R}}$$

Lemma 3.3 $x \in \mathbb{R} \neq 0, \hat{\eta}(x) = (\hat{s}, \hat{e}, \hat{f}), \hat{r}$

1. $\hat{r}(x) = \hat{(}r)([x]_{p-\hat{e}})$
2. $\hat{\eta}([x]_{p-\hat{e}}) = (\hat{s}, \hat{e}, [\hat{f}]_p)$
3. $x' =_{p-\hat{e}} x \Rightarrow \hat{r}(x) = \hat{r}(x')$

3.3 How to compute $r(x)$?

1. $\eta\{x\} = (s, e, f)$ IEEE-norm shift
2. $f_1 = signrd(s, f) \in [0, 2]$ significant rounding depending on mode
3. $(e_2, f_2) = post(e, f_1) = \begin{cases} (e+1, \frac{f}{2}) & f_1 = 2 \\ (e, f_1) & otherwise \end{cases}$
4. $(s, e_3, f_3) = exprd(s, e_2, f_2)$ exponent round depending on mode

Lemma 3.4 (Decomposition Theorem of Rounding)

$$(s, e_3, f_3) = \eta(r(x))$$

Proof: $F = \{\langle q[0].q[1:p-1]\rangle | q[i] \in \{0, 1\}\} \cup \{2\}$
$Y' = (min\{x \in F | f \leq x\});\ Y = (min\{x \in F | f \geq x\})$
$signrd_z(s, f) = Y(f)$
$signrd_u(s, f) = \begin{cases} Y'(f) & : s = 0 \\ Y(f) & : s = 1 \end{cases}$
$signrd_d(s, f) = \begin{cases} Y(f) & : s = 0 \\ Y'(f) & : s = 1 \end{cases}$
$signrd_ne(s, f) = \begin{cases} Y'(f) & : f < f' \ or \ f = f' \wedge f[p-1] = 0 \\ Y(f) & : f > f' \ or \ f = f' \wedge f[p-1] = 1 \end{cases}$
$X_1 = [\![s, e, f_1]\!] = (-1)^s * 2^e * f_1$

Lemma 3.5 $X_1 = \begin{cases} r(x) & : |X| \leq X_{max} \\ \hat{r}(x) & : |X| > X_{max} \end{cases}$

Proof: 4 cases: $s = 0, 1$, f normal/denormal

- $s = 0$: f normal: $f \in [1, 2) x \in [2^e, 2^{e+1}]$
- $s = 1$: f normal: mirror s=0
- denormal: similar picture

Remnark: $r(x) = \hat{r}(x)$ if $|x| \leq X_{max}$ and f is normal.

Lemma 3.6 *$[(s, e_2, f_2) = \eta(r(x))]$*

Proof: $x_1 = x_2 = [\![s, e_2, f_2]\!]$
to show: (s, e_2, f_2) is IEEE normal, i. e.
f_2 normal if $|X| \geq 2^{e_{min}}$
f_2 denormal if $|X| < 2^{e_{min}}$ and $e = e_{min}$
$|X| \geq 2^{e_{min}} \Rightarrow f_1 \in [1 : 2]$

- $f_1 \in [1 : 2) \Rightarrow f_2 = f_1$ normal
- $f_1 = 2 \Rightarrow f_2 = 1$ normal

$|X| < 2^{e_{min}} \Rightarrow f_1 \in [0 : 1]$

- $f_1 \in [0 : 1) \Rightarrow f_2 = f_1$ denormal
- $f_1 = 1 \Rightarrow f_2 = f_1$ normal

exponent rounding

- $exprd_u(s, e_2, f_2) = \begin{cases} (\infty, 0) & : e_2 > e_{max} \wedge (s = 0) \\ (e_{max}, 2 - 2^{-(p-1)}) & : e_2 > e_{max} \wedge (s = 1) \\ (e_2, f_2) & : otherwise \end{cases}$
- $exprd_d(s, e_2, f_2) =$ similar
- $exprd_z(s, e_2, f_2) =$ similar
- $exprd_{ne}(s, e_2, f_2) = \begin{cases} (\infty, 0) & : e_2 > e_{max} \\ (e_2, f_2) & : otherwise \end{cases}$

$$x_3 = [\![s, e_3, f_3]\!]$$

show $(s, e_3, f_3) = \eta(r(x))$
easy case: $exprd(s, e_2, f_2) = (e_2, f_2)$ IEEE normal and $x_3 = r(x)$
other case: $exprd(s, e_2, f_2) \neq (e_2, f_2)$, also IEEE normal since $(\infty, 0)$ IEEE normal

remains to be shown:

$$x_3 = r(x)$$

easy case: $|x| \leq X_{max} \Rightarrow x_2 = x_1 = r(x)(L\ 3.6)$
$(s, e_2, f_2) = \eta(r(x)) \Rightarrow e_2 \leq e_{max} \Rightarrow (e_3, f_3) = (e_2, f_2)$
case $|x| > X_{max}$:
For all rounding modes $\hat{r}(x) \neq r(x) \Leftrightarrow \hat{r}(x) > X_{max}$
$\Leftrightarrow \hat{r}(x) \geq 2^{e_{max}+1} \Leftrightarrow e_2 > e_{max}$
for $r = r_u : x_3 = [\![s, e_3, f_3]\!]$

$$= \begin{cases} \infty & : e_2 > e_{max} \wedge (s = 0) \\ -X_{max} & : e_2 > e_{max} \wedge (s = 1) \\ x_2 & e_2 \leq e_{max} \end{cases}$$

$$= \begin{cases} \infty & : \hat{r}(x) \neq r(x) \wedge (s = 0) \\ -X_{max} & : \hat{r}(x) \neq r(x) \wedge (s = 1) \\ x_2 & e_2 \leq e_{max} \end{cases}$$

$= r(x)$

Algorithm $\eta(r(x)) = (s, exprd(s, post(e, sigrd(s, f))))$
$x \neq 0 :\ (s, \hat{e}, \hat{f}) = \hat{\eta}(x)$
$\hat{\eta}(\hat{r}(x)) = (s, post(\hat{e}, sigrd(s, \hat{f})))$

3.4 Exception

INV	invalid operation	$e[i]$
DBZ	devision by zero	$e[i+1]$
OVF	overflow	$e[i+2]$
UNF	underflow	$e[i+3]$
INX	inexact	$e[i+4]$

Table 1: Interrupt Handling (Architecture 1)

- IEEEf: interrupts are internal
- $ev[i]$ i'th interrupt signal
- $SR[i]$ bit i of Status Register
  mask bit for $eev[i] \geq 5$
- $MCA[i] = \begin{cases} ev[i] & i \leq 4 \\ ev[i] \wedge SR[i] & i \geq 5 \end{cases}$
  Masked Cause
- $JISR = \bigvee_i MCA[i]$
  Jump for Interrupt Service Routine

Case 1: $SR[j] = 0 :\ j = i, \ldots, i+4$ all IEEE interrupts disabled
$ev[j]$ are accumulated in special purpose register:

IEEE	extra result and operant
SR	extra operant
RM	extra operant
IEEE flags	relevant for FP comp.
status register	relevant for FP comp.
rounding mode	relevant for FP comp.

OVF, UNF, INX: def. and computation with representatives.
$x = a \circ b,\ a, b \in \mathbb{Z}$

$OVF(x) \leftrightarrow |\hat{r}(x)| > X_{max}$
$\hat{\eta}(x) = (s, \hat{e}), \hat{f})$
$\hat{r}([x]_{p-\hat{e}}) = \hat{r}(x)$
$OVF(x) \leftrightarrow OVF([x]_{p-\hat{e}})$

$OVF(x) \leftrightarrow 2^{\hat{e}} * sigrd(\hat{f}) > X_{max}$
$\leftrightarrow (\hat{e} > e_{max})$ OVF before rounding
$\vee(\hat{e} = e_{max}) \wedge (sigrd(\hat{f}) = 2)$ OVF after rounding

$UNF(x) \leftrightarrow Tiny(x) \wedge Loss(x)$

$Tiny_a(x) \leftrightarrow 0 < |\hat{r}(x)| < 2^{e_{min}}$ Tiny after rounding
$Tiny_b(x) \leftrightarrow 0 < |x| < 2^{e_{min}}$ Tiny before rounding
$Tiny_a(x) \rightarrow Tiny_b(x)$
$\hat{\eta}(x) = (s, \hat{e}, \hat{f})\ Tiny_b(x) \leftrightarrow Tiny_b([x]_{p-\hat{e}})$
$\hat{r}(x) = \hat{r}([x]_{p-\hat{e}}) \rightarrow Tiny_a(x) = Tiny_a([x]_{p-\hat{e}})$

$Loss_a(x) \leftrightarrow r(x) \neq \hat{r}(x)$ denormalization loss
$Loss_b(x) \leftrightarrow r(x) \neq x$ inexact
$\hat{\eta}(x) = (s, \hat{e}, \hat{f})\ [x]_{p-\hat{e}} =_{p-\hat{e}} x$
$\hat{e} \leq e \Rightarrow [x]_{p-\hat{e}} =_{p-\hat{e}} x$
$r([x]_{p-\hat{e}}) = r(x)$

$Loss_a(x) \rightarrow Loss_b(x)$
proof: assume false $r(x) = x \Rightarrow \hat{r}(x) = x$
exercise: $Loss_b(x) \leftrightarrow Loss([x]_{p-\hat{e}})$
$X_{max} > |x| \geq 2^{e_{min}} \rightarrow r(x) = \hat{r}(x)$
$|x| < 2^{e_{min}}$
$r(x) = x \leftrightarrow x \in mathcalR,\ x$ denormal

$r([x]_{p-\hat{e}}) = [x]_{p-\hat{e}} \leftrightarrow [x]_{p-\hat{e}} \in \mathcal{R}$, denormal

proof: $Loss_b(x) \rightarrow Loss_b([x]_{p\hat{e}})$
$Loss_b(x) \leftrightarrow r(x) \neq x,\ Loss_b([x]_{p\hat{e}}) = r([x]_{p\hat{e}}) \neq [x]_{p\hat{e}}$
$r([x]_{p\hat{e}}) = r(x),\ [x]_{p\hat{e}} =_{p\hat{e}} x$
$Loss_b([x]_{p\hat{e}}) \rightarrow Loss_b x$
$Loss_b([x]_{p\hat{e}}) \leftrightarrow r([x]_{p\hat{e}}) \neq [x]_{p\hat{e}}$
$Loss_b(x) \leftrightarrow r(x) \neq x$
$r([x]_{p\hat{e}}) = r(x),\ [x]_{p\hat{e}} =_{p\hat{e}} x$

Wrapped exponent Let $\alpha = 3 * 2^{n-1}$
$x = a \circ b, a, b \in \mathcal{R}, \circ \in \{+, -, *, /\}$
$a \circ b = r(y)\ y = \begin{cases} x * 2^{-\alpha} & : OVF \wedge OVFen \\ x * 2^{\alpha} & : UNF \wedge UNFen \end{cases}$

$\left.\begin{array}{l} OVF(x) \rightarrow 2^{e_{min}} < x * 2^{-\alpha} < X_{max} \\ UNF(x) \rightarrow 2^{e_{min}} < x * 2^{\alpha} < X_{max} \end{array}\right\} \begin{array}{l} \in \mathcal{R} \\ \text{r(y) representable if } OVF \wedge OVFen \\ \vee UNF \wedge UNFen \end{array}$

proof: for $* OVF$
Largest result:
$X_{max}^2 < (2^{e_max+1})^2 = 2^{e_max+2}$
$2 * e_{max} + 2 - \alpha = 2(2^{n-1} - 1) + 2 - 3 * 2^{n-2} = 2^{n-2} < e_{max}$
no overflow unless $|x| > X_{max} > 2^{e_{max}}$
$e_{max} - \alpha = 2^{n-1} - 1 - 3 * 2^{n-2} = -2^{n-2} - 1 > -2^{n-1} + 2 = e_{min}$

$\hat{\eta}(\hat{r}(x)) = (s, u, v) \Rightarrow$

1. $OVF(x) \rightarrow \eta(x * 2^{-\alpha}) = (s, u - \alpha, v)$
2. $UNF(x) \rightarrow \eta(x * 2^{\alpha}) = (s, u + \alpha, v)$

proof 1: $\hat{\eta}(x) = (s, \hat{e}, \hat{f}) \Rightarrow \hat{\eta}(x * 2^{-\alpha}) = (s, \hat{e} + \alpha, \hat{f})$
Let $f_1 = sigrd(s, [\hat{f}]_{p-1\hat{e}})$ and $(u, v) = post(\hat{e}, f_1)$
$\Rightarrow (u - \alpha, v) = post(\hat{e} - \alpha, f_1)$
$\Rightarrow \hat{\eta}(\hat{r}(x * 2^{-\alpha})) = (s, post(\hat{e} - \alpha, f_1)) = (s, u - \alpha, v)$ decomposition theorem

$INX(y) \leftrightarrow (r(y) \neq y) \vee (OVF(y) \wedge \overline{OVFen})$ inexact result
Y finite $INX(y) \leftrightarrow r(y) \neq y$
$\eta(x) = (s, e, f),\ \hat{\eta}(x) = (s, \hat{e}, \hat{f})$

$OVF(x) \wedge OVFen \vee UNF(x) \wedge UNFen$
$\Rightarrow INX(y) \leftrightarrow sigrd(s, \hat{f}) \neq \hat{f}$
$\leftrightarrow sigrd(s, [\hat{f}]_p) \neq [\hat{f}]_p$
otherwise $\Rightarrow INX(y) \leftrightarrow sigrd(s, f) \neq f \vee OVF(x)$
$\leftrightarrow sigrd(s, [f]_p) \neq [f]_p \vee OVF([x]_{p-e})$

4 Auxiliary circuits

4.1 shifter

Definition 4.1 ($cls(a,x),\ crs(a,x)$**)** $a \in \{0,1\}^n,\ n = 2^m,\ x \in [0 : n-1]$

$$cls(a,x) = (a[n-1-x:0]a[n-1:n-x])$$
$$crs(a,x) = (a[x-1:0]a[n-1:x])$$

n-CLS $c = cls(a, < b >)$

Figure 32: n-CLS

(n,x)-CLS $c = \begin{cases} cls(a,x) & : s = 1 \\ a & : s = 0 \end{cases}$

Figure 33: (n,x)-CLS

Figure 34: CLS circuit

Correctness: Ind(i): $a^{(i)} = cls(a, \langle b[i:0] \rangle)$

Lemma 4.1 (crs(a,x))

$$crs(a,x) = cls(a, n - x \ mod \ n)$$

n-lr-shifter $c = \begin{cases} cls(a, \langle b \rangle) & : r = 0 \\ crs(a, \langle b \rangle) & : r = 1 \end{cases}$

$n - \langle b \rangle = n + \overline{b} + 1 \ mod \ n = \langle \overline{b} \rangle + 1 \ mod \ n$

4.2 Half-decoder

$Y[2^n - 1 : 0] = 0^{2^n - \langle x \rangle} 1^{\langle x \rangle}$
$n = 1:$
$n > 1$
$L = [2^{n-1} - 1 : 0]$ lower bits, $H = [2^n - 1 : 2^{n-1}]$ higher bits

Figure 35: n-lr-shifter

Figure 36: n-bit half-decoder symbol

Correctness: $X_{n-1} = 0 \Rightarrow \langle x \rangle = \langle x[n-2:0] \rangle$
$Y[H] = 0^{2^{n-1}}\ Y[L] = V[L]$

$X_{n-1} = 1 \Rightarrow \langle x \rangle = 2^{n-1} + \langle x[n-2:0] \rangle$
$Y[H] = V[L]\ Y[L] = 1^{2^{n-1}}$

4.3 Leading Zero Counter

$x[n-1:0],\ n = 2^m,\ H = [n-1:\frac{n}{2}],\ L = [\frac{n}{2}-1:0]$
$lz(x) = \langle y \rangle,\ y = y[m:0]$

$y_H = lz(x[H]),\ y_L = lz(x[L])$

$$lz(x[H]x[L]) = \begin{cases} lz(x[H]) & : lz(x[H]) < 2^{n-1} \\ 2^{m-1} + lz(x[L]) & : otherwise \end{cases}$$
$$= \begin{cases} \langle 0 y_H \rangle : y_H[m-1] = 0 \\ \langle 10^{m-1} \rangle + \langle y_L[m-1:0] \rangle : y_H[m-1] = 1 \end{cases}$$
$$= \begin{cases} \langle 0 y_H \rangle : y_H[m-1] = 0 \\ \langle 01 y_L[m-2:0] \rangle : y_H[m-1] = 1 \wedge y_L[m-1] = 0 \\ \langle 10 y_L[m-2:0] \rangle : y_H[m-1] = 1 \wedge y_L[m-1] = 1 \end{cases}$$

5 FPU-Overview

Fig. 8.1
db: double precision
$a = [\![s_a, e_a[n-1:0], f_a[1:p-1]]\!],\ b = [\![s_b, e_b[n-1:0], f_b[1:p-1]]\!]$
$$(p, n) = \begin{cases} (53, 11) & : db \\ (24, 8) & : \overline{db} \end{cases}$$
single precision is embedded at the high end of a 64-bit vector.
$(s_a, e_a[n-1:0], f_e[1:p-1])$
$$= \begin{cases} (FA2[63], FA2[62:55], FA2[54:32]) & : \overline{db} \\ (FA2[63], FA2[62:52], FA2[51:30]) & : db \end{cases}$$

outputs of the unpacker:
$$a = \begin{cases} (-1)^{s_a} * 2^{[e_a[10:0]]} * \langle f_a[0].f_a[1:p-1] \rangle & : \overline{normalize} \\ (-1)^{s_a} * 2^{[e_a[10:0]] - lz_a[5:0]} * \langle f_a[0].f_a[1:p-1] \rangle & : normalize \end{cases}$$

Figure 37: 1-bit half-decoder

Figure 38: n-bit half-decoder

$$x = a \circ b,\ \hat{\eta}(x) = (s, \hat{e}, \hat{f})$$

$$[\![s_r, e_r, f_r]\!] =_{p-\hat{e}} x$$

$e_r[12:0],\ fr[-1, 0, 1:55]$
$f_r[-1:0] = 00 \Rightarrow \overline{OVF(x)}$

$$a = [\![s_a, e_a[n-1:0], f_a[1:p-1]]\!]$$
$$= \begin{cases} (-1)^{s_a} * 2^{[\![e_a]\!]_{bias}} * \langle 1.f_a \rangle & : e_a \notin \{0^n, 1^n\} \\ (-1)^{s_a} * 2^{e_{min}} * \langle 1.f_a \rangle & : e_a = 0^n \\ (-1)^{s_a} * \infty & : e_a = 1^n \wedge f = 0^{p-1} \\ NAN & : e_a = 1^n \wedge f \neq 0^{p-1} \end{cases}$$

unpacker spec:

$$a = \begin{cases} (-1)^{s_a} * 2^{[e_a[10:0]]} * \langle f_a[0].f_a[1:p-1] \rangle & : normal \\ (-1)^{s_a} * 2^{[e_a[10:0]] - \langle lz[5:0] \rangle} * \langle f_a[0].f_a[1:p-1] \rangle & : \overline{normal} \end{cases}$$
$$normal = (f_a[0] = 1)$$
$$e_z = 1 \leftrightarrow e_a = \begin{cases} 0^{11} & : db \\ 0^8 & : \overline{db} \end{cases} \quad e_{inf} = 1 \leftrightarrow e_a = \begin{cases} 1^{11} & : db \\ 1^8 & : \overline{db} \end{cases}$$
$$Incin = \begin{cases} e_a & : e_a \neq 0^n \\ e_{min} & : e_a = 0^n \end{cases}$$
$$[e[10:0]] = \begin{cases} [\![e_a]\!]_{bias} & : e_a \neq 0^n \\ e_{min} & : e_a = 0^n \end{cases}$$
$$\langle 0.h[1:52] \rangle = \langle 0.f_a[1:p-1] \rangle\ p \in \{53, 24\}$$
$$= \begin{cases} \langle 0.f_a[1:52] \rangle & : db \\ \langle 0.f_a[1:23] \rangle & : \overline{db} \end{cases}$$
$$h[0] = \begin{cases} 0 & : e_a = 0^n \\ 1 & : e_a \neq 0^n \end{cases} \quad \text{hidden bit } f_a[0] = h[0]$$
$$f_z = 1 \leftrightarrow f[1:p-1] = 0^{p-1}$$

Übung CLS(53)
$norm = 0$ trivial
$norm = 1$
$$a = (-1)^{s_a} * 2^{[e_a] - \langle lz[5:0] \rangle} * \langle f_a[0].f_a[1:52] \rangle$$
$$\langle h \rangle = \begin{cases} \langle f_a \rangle & : \overline{normal} \\ \langle f_a \rangle - 2^{-\langle lz \rangle} & : normal \end{cases}$$

Figure 39: n-bit leading zero counter

Figure 40: CLS(53)

6 Adder/Subtracter

$a = [\![s_a, e_a, f_a]\!]$
$b = [\![s_b, e_b, f_b]\!]$
$(s_a, e_a, f_a) = \eta(a)$
wlog $\delta = (e_a - e_b) \geq 0$

$$\begin{aligned} S &= [\![s_a, e_a, f_a]\!] + [\![s_b, e_b, f_b]\!] \\ &= (-1)^{s_a} * 2^{e_a} * f_a + (-1)^{s_b} * 2^{e_a} * f_a \\ &= 2^{e_a} * ((-1)^{s_a} * f_a + (-1)^{s_b} * 2^{-\delta} * f_b) \\ &= 2^{e_a} * ((-1)^{s_a} * f_a + (-1)^{s_b} * f') \end{aligned}$$

where $f' = [2^{-\delta} * f_b]_{p+1}$

Assume $s \neq 0$ $\quad \hat{\eta}(S) = (s, \hat{e}, \hat{f})$
$S =_{\hat{e}-p} x \Rightarrow r(S) = r(x)$
$S =_{p-\hat{e}} 2^{e_a} * ((-1)^{s_a} * f_a + (-1)^{s_b} * f')$

case $\delta \leq 3 : f' = 2^{-\delta} * f_b \Rightarrow$ exact result
case $\delta \geq 2 : 2^{-\delta} * f_b < 2^{-2} * 2 = 1/2$

$e_a, e_b \geq e_{min}$
$e_a = e_b + \delta \geq e_{min} + 2 \rightarrow f_a$ is normal
$\|(-1)^{s_a} * f_a + (-1)^{s_b} * f_b\| > 1/2 \Rightarrow \hat{e} \geq e_a - 1 > e_{min}$
$f_a \geq 1, f_b < 1/2\ p - \hat{e} \leq p - e_a + 1$
$(-1)^{s_a} * f_a + (-1)^{s_b} * 2^{-\delta} * f_b =_{p+1} (-1)^{s_a} * f_a + (-1)^{s_b} * f'$
$S =_{p+1-e_a} 2^{e_a} * ((-1)^{s_a} * f_a + (-1)^{s_b} * f')$
$S =_{p-\hat{e}} 2^{e_a} * ((-1)^{s_a} * f_a + (-1)^{s_b} * f')$,since $p - \hat{e} \leq p - e_a + 1$

$sb' = sb \oplus sub$ effective substraction
$e_b\_gt\_e_a \leftrightarrow [e_b] > [e_a] \leftrightarrow [e_a] - [e_b] < 0$ sign extension

after wrap shift f_b right
unlimited shift distance: $|\delta|$ $\quad \delta = [e_a] - [e_b] = [a_{s1}]$

$$a_{s1}[n-1:0] = \begin{cases} a_{s1}[n-1:0] : a_s \geq 0 \\ \overline{a_{s1}[n-1:0]} : a_s < 0 \end{cases}$$

$a_s \geq 0 \rightarrow \langle a_{s1} \rangle = \delta$
$a_s < 0 \leftarrow \delta - 1 = -[a_s[n:0]] - 1 = \overline{[a_s[n:0]]}$
$0 \leq |\delta| - 1 \leq 2^n - 1 \Rightarrow \overline{a_s[n]} = 0$
$\overline{\langle a_s[n-1:0] \rangle} = [a_s[n:0]] = |\delta| - 1$

$$\langle a_{s1} \rangle = \begin{cases} |\delta| : e_a \geq a_b \\ |\delta| - 1 : e_a < e_b \end{cases}$$

compensation of approximation error
preshift a-operand in case of wrap

$$\langle (f_{a_2}, f_{b_2} \rangle = \begin{cases} (f_a, f_b) : e_a \geq a_b \\ (f_b, f_a/2 : e_a < e_b \end{cases}$$

$$2^{\langle a_{s_1} \rangle} * f_{b_2} = \begin{cases} 2^{-\delta} * f_b : e_a \geq a_b \\ 2^{-\delta} * f_a : e_a < e_b \end{cases}$$

$f_{b_3} = [2^{-\langle a_s \rangle} * f_{b_2}]_{p+1}$
limiting shift distance
$b = \lceil log(p+3) \rceil$
$B = 2^b - 1 = \langle 1^b \rangle \geq p + 2$
$\langle a_{s_1} \rangle \geq B \leftrightarrow \bigvee_{i=b}^{n-1} a_{s1}[i] = 1$

$$\langle a_{s2} \rangle = \begin{cases} B & : \langle a_{s1} \rangle \geq B \\ \langle a_{s1} \rangle & : otherwise \end{cases}$$

$$sticky = \vee_{j=p+2-\langle a_{s2} \rangle}^{p+1} f_{b_2}[j]$$

$f_{b_2}[0].f_{b_2}[1:53] f_{b_2}[54]$
last $\langle a_{s2} \rangle$ bits of $f_b[...54]$
feed $\langle a_{s2} \rangle$ into half-decoder

$$S =_{p-\hat{e}} \begin{cases} 2^{e_a} * ((-1)^{s_a} * f_a + (-1)^{s'_b} * [2^{-\delta} * f_b]_{p+1}) & : fgd \\ 2^{e_a} * ((-1)^{s_a} * f_a + (-1)^{s'_b} * [2^{-\delta} * f_b]_{p+1}) & : fgd \end{cases}$$

$s'_b = s_b \oplus sub$ subtract b
$s_x = s_a \oplus s'_b$ subtract significant

$$\begin{aligned} sum &= f_{a_2} + (-1)^{s_x} * f_{b3} \\ &= \langle f_{a_2}[0].f_{a_2}[1:p-1] \rangle + (-1)^{s_x} \langle f_{b_3}[0].f_{b_3}[1:p+2] \rangle \\ &= [0^2 f_{a_2}[0] - f_{a_2}[1:p-1]0^3] + [s_x^2 f_{b_3}[0] \oplus s_x f_{b_3}[1:p+2] \oplus s_x] + s_x 2^{-(p+2)} \\ &= sum[-2:0].sum[1:p+2] \end{aligned}$$

$(0 \rightarrow 0^2) \wedge (s_x \rightarrow s_x^2) \Rightarrow$ no overflow

6.1 special cases:

$INV = SNAN_a \vee SNAN_b \vee (INF_a \wedge INF_b \wedge s_x)$
$NAN = IVV \vee NAN_a \vee NAN_b$
$INF = (INF_a \vee INF_b) \wedge \overline{NAN_b}$

$$S_{s_3} = \begin{cases} s_a & : INF_a \\ s'_b & : INF_b \wedge \overline{INF_a} \end{cases}$$

sign bit for zero result
$r_d = RM[1:0] = 11$ rounding mode coding
$result = -0$ if $(f_s = 0) \wedge s_x$

$$S_{s_2} = \begin{cases} 0 & : s_x \wedge (RM[1:0] \neq 11) \\ 1 & : s_x \wedge (RM[1:0] = 11) \\ s_a & : \overline{s_x} \end{cases}$$
$$S_s = \begin{cases} S_s 3 & : INFS \\ S_s 2 & : \neg INFS \wedge (f_s = 0) \\ S_s 1 & : \neg INFS \wedge (f_s \neq 0) \end{cases}$$

7 FP-MULT-DIV

unpacker: norm -1
$\hat{\eta}(a) = (s_a, e_a - lza, f_a)$
$\hat{\eta}(b) = (s_b, e_b - lza, f_b)$

$s_q = s_a \oplus s_b$
$e_q = e_a - lza - (e_b - lza)$
$q = f_a / f_b \in (1/2, 2)$
$a/b = [\![s_q, e_q, q]\!]$
$\hat{\eta}(a/b) = (s_q, \hat{e}, \hat{q})$

Let $f_d = [q]_{p+1}$
$2^{e_q} f_d =_{p+1-e_q} 2^{e_q} * q$
$\Rightarrow 2^{e_q} f_d =_{p-\hat{e}} 2^{e_q} q$

computing $f_d(s_q, e_q, f_d) \rightarrow$Rounder

- x_0 is approximation of $1/f_b$ from ROM
- i is Newton Iteration: x_i approximation if $1/f_d$
- compare $b * x_i$ with $\rightarrow f_d$

$$x_i = x_{i+1} = f(x_i)/f'(x_i)$$

Figure 41: newton iteration

$$x_{i+1} = x_i + f(x_i)/f'(x_i)$$

Inverse_Computation for f_b

$f(x) = 1/x - f_b \qquad f'(x) = -1/x^2$

$x_{i+1} = x_i + (1/x_i - f_b)x_i^2$
$\quad = x_i(2 - f_b x_i)$

Error $\delta_i = 1/f_b - x_{i+1}$

$\delta_{i+1} = 1/f_b - x_{i+1}$
$\quad = 1/f_b - x_i(2 - f_b x_i)$
$\quad = 1/f_b - 2x_i + f_b x_i^2$
$\quad = f_b(1/f_b + x_i)^2$
$\quad = f_b \delta_i < 2\delta_i^2$

$f_b = \langle 1.f_b[1 : p-1]\rangle \in [1, 2)$

Figure 42: newton iteration 2

$f_b' = \langle 1.f : b[1 : \gamma]1\rangle \qquad x' = 1/f_b'$

$f_b \neq 1 : x_0)r_0(x')$
$f_b = 1$: special hardware forward f_a

Romsize: $2^\gamma\gamma$
$f_b' = \langle 1f_b[1 : \gamma]1\rangle$ address
$x' = \langle 0.1x'[2 : y+1]\rangle$result

$$0 < |\delta_0| = |1/\delta_b - x_0| < 1.5 * 2^{-(\gamma+1)}$$

upper bound
$|f(u) - f_(v)| \leq |v - u| * |f'(u)|$
$|f_b - f_{b'}| \leq 2^{-(\gamma-1)}$
$|1/f_b - 1/f_{b'}| \leq 2^{-(\gamma-1)}$
$|x' - x_0| \leq 1/2 * 2^{-(\gamma+1)}$

Notation: $\lfloor f \rfloor_\sigma \lfloor f2^\sigma \rfloor /2^\sigma$

$f = \langle f[-a:0] : f[1:b] >$
σ truncate after bit σ in hardware
$\lfloor f \rfloor_\sigma = \langle f[-a:0] : f[1:\sigma] \rangle$

$$x_{i+1} = \lfloor x_i \lfloor 2 - f_i x_i \rfloor_\sigma \rfloor_\sigma$$

$z = f_b x_i \qquad z = \langle z[0].z[1:\delta] \rangle$

Lemma 7.1 $0 < 2 - z \leq \langle z[0].z[1:\delta] \rangle + 2^{-\delta}$

$$\begin{aligned} 2 - z &= \langle 10.0^s \rangle - \langle < 0 z[0].z[1:\delta] \rangle \\ &= \langle 10.0^s \rangle + \langle 1 z[0].z[1:s] + 2^{-s} \bmod 4 \\ &= \langle \overline{z[0].z[1:s]} \rangle + 2^{-s} \\ &= \langle \overline{z[0].z[1:\sigma]} \rangle + 2^{-s} + \textstyle\sum_{i=\sigma+1}^{s} \overline{z[i]} * 2^{-i} \\ &= \langle \overline{z[0].z[1:\sigma]} \rangle + 2^{-\sigma} \end{aligned}$$

$z_i = f_b x_i$
$A_i = \langle z_i[0].z_i[1:\delta] \rangle$
$x_i + 1 = \lfloor A_i x_i \rfloor_\sigma$

Lemma 7.2 $\sigma \geq 4 \qquad x_o \in (1/2, 1) \qquad 0 \leq |\delta_o| < 1/8$
$\Rightarrow x_i + 1 \in (0, 1)$

$0 < \delta_{i+1} < 2\delta_i^2 + 2^- \sigma + 1 < 1/4$
$\delta_{i+1} = \Delta_1 + \Delta_2 + \Delta_3$
$\Delta_1 = 1/f_b - x_i(2 - z_i)$
$\Delta_2 = x_i(2 - z_i) - x_i A_i$
$\Delta_3 = x_i A_i - \lfloor x_i A_i \rfloor_\sigma$

$0 < \Delta_1 < 2\delta_i^2$
$x_i \in (0, 1) \qquad 0 < z_i = f_b x_i < 2 \qquad z_i \in (0, 2)$
$0 < \Delta_2 = x_i(2 - z_i - A_i) \leq x_i 2^{-\sigma} < 2^- \sigma$
$0 \leq \Delta_3 \leq 2^{-\sigma}$
$\delta_{i+1} < 2\delta_i^2 + w^{-\sigma+1} < 1/8 + 1/8 = 1/4$
$0 < \sigma_{i+1} = 1/f_b - x_{i+1} < 1/4$
$\Rightarrow 1/4 \leq 1/f_b - 1/4 < x_{i+1} < 1/f_b \leq 1$

Lookup Table: $2^\gamma * \gamma$ RAM
error: $\leq 1.5 * 2^{-(\gamma+1)}$

Lemma 7.3 *choose* $\sigma = 57; \gamma = 8$

$$i = \begin{cases} 2 & : p = 24 \\ 3 & : p = 53 \end{cases}$$
$\Rightarrow \delta_i < 2^{-(p+2)}$

Proof: $\delta_0 < 1.5 * 2^{-9}$
$\delta_1 < 2 * (1.5)^2 2^{-18} + 2^{-56} < 4.6 * 2^{-18}$
$\delta_2 < 42.32 * 2^{-36} + 2^{-56} \leq 42.33 * 2^{-36} < 2^3 0$
$\Rightarrow i = 2$ iteration suffice for p = 24
$\delta_3 < 3583.7 * 2^{-71} + 2^{-51}$
$\leq 3.5 * 2^{-62} + 2^{-56} < 2^{-55}$
$\Rightarrow i = 3$ iteration suffice for p = 53

$0 < 1/f_b - x_i < 2^{-(p+2)}$
$x_i < 1/f_b < x_i + 2^{-(p+2)}$
$f_a x_i < f_a/f_b = q < f_a x_i + 2^{-(p+1)} < \lfloor f_a x_i \rfloor_{p+1} + 2^{-p}$
$E = \lfloor f_a x_i \rfloor_{p+1}$

Figure 43: E to $E + 2^{-p}$

$$\lfloor f_a/f_b \rfloor_{p+1} = \begin{cases} E + 2^{-(p+2)} & : f_a/f_b < E' \\ E + 2^{-(p+1)} & : f_a/f_b = E' \\ E + 3 * 2^{-(p+2)} & : f_a/f_b > E' \end{cases}$$

$\circ \in \{<, =, >\}$
$f_a/f_b \circ E' \leftrightarrow f_a \circ f_b * E'$

$a = (-1)^{s_a} 2^{e_a - lz_a} fa$
$ea = \langle ea[10:0] \rangle \qquad \langle lz[5:0] \rangle \qquad \langle fa[0].fa[1:52] \rangle$

$x = a \circ b \qquad \circ \in \{*, /\}$
$\hat{\eta}(x) = (s, \hat{e}, \hat{f})$ compute
$(s_q, e_q, f_q) = [\![s_q, e_q, f_q]\!] =_{p-\hat{e}} x$
$s_q = s_a \oplus s_b$
$$e_q = \begin{cases} e_a - lz_a + (e_b - lz_b) & : \neg div \\ e_a - lz_a - (e_b - Lz_b) & : div \end{cases}$$
$2^n - 2 = 2e_{max} \geq e_q \geq -2e_{min} - p > -2^{n+1}$
$n = 11 \Rightarrow$ use 13 bit arithmetic
$f_a, f_b \in [1, 2)$
$f_a * f_b \in [1, 4)$
$\hat{e} \geq e_q$

significand multiplication
$\langle f_m[-1:55]\rangle = [\langle f_a[0].f_a[1:52]0^5\rangle * \langle f_b[0].f_b[1:52]2^5\rangle]_{54}$
$[\![s_q, e_q, f_q]\!] = (-1)^{s_q}2^{e_q}[f_a f_b]_{54} =_{54-\hat{e}} (-1)^{s_q}2^{e_q}(f_a f_b)$

Figure 44: FSD underlying the iterative Division

$Newton\ 1,2 : Dcnt--; A = appr(2 - x * f_b)$
$Newton\ 3,4 : x = \lfloor x * A \rfloor_{57}$
$Quotient\ 1,2 : E = \lfloor a * x \rfloor_{p+1}$
$Quotient\ 3,4 : E_b = E * f_b$

after lookup:
$x = x_0$
$dcnt = \begin{cases} 2 & : \overline{db} \\ 3 & : db \end{cases}$
after i'th state:
$A = A_{i-1}$
$x = x_i$
$dcnt = dcount - i$

$E = \lfloor f_a x_i \rfloor_{p+1} \in (0,2)$ Quotient 1-4)
$E_b = E * f_b \qquad D_a = f_a \qquad D_b = f_b$
$f_a = \begin{cases} E + 2^{-(p+2)} \\ E + 2^{-(p+2)} \\ E + 1.5 * 2^{-(p+2)} \end{cases}$
$E' = E + 2^{-(p+1)} \qquad p \in 24, 53$

$db = 1:$
$\langle E'[-1:0].E'[1:54]\rangle = \langle E[0].E[1:54]\rangle + 2^{-54}$

$db = 0:$
$\langle E'[-1:0].E'[1:25]\rangle = \langle E[0].E[1:25]\rangle + 2^{-25}$
$= \langle E[0].E[1:25]1^{25}\rangle + 2^{-54}$

Computation of β
$f_{sb} = f_b * 2^{-(p+1)}$ shift D_b right by $p+1$ bits

$\langle 0.0^{24} f_{sb}[25:106]\rangle = \begin{cases} \langle 0.0^{24}0^{29}D_b[0:52]\rangle & : db \\ \rangle 0.0.^{24}D_b[0:52]0^{25}\rangle & : \neg db \end{cases}$

$\beta = f_a - E_b - f_b * 2^{-p+1}$
$= [00D_a[0].D_a[1:52]0^{54}]$
$-[0E_b[-1:0].E_b[1:106]]$

$$
\begin{aligned}
&-[000.0^{24} f_{sb}[25:106] \\
&= [00 D_a[0].D_a[1:52] 0^{54}] \\
&\quad +[1\overline{E_b[-1:0]}.\overline{E_b[1:106]}] \\
&\quad +[1^3.1^{24}\overline{f_{sb}[25:106]} + 2^{-105}
\end{aligned}
$$

note: $[00 D_a[0].D_a[1:52] o^{54}] + 2^{-105}$
$= [00 D_a[0].D_a[[1:52]0^{52}10]$

Figure 45: 108-3/2-add

7.1 FP-Rounder

$x \in R \backslash \{0\}$

$$
y = \begin{cases} x * 2^{-\alpha} & : OVF \wedge OVFen \\ x * 2^{\alpha} & : UNF \wedge UNFen \\ x & : otherwise \end{cases}
$$

$\hat{\eta}(x) = (s, \hat{e}, \hat{f})$
$\hat{\eta}(x) = (s, e, f)$
result $[\![s, e_{out}, f_{out}]\!] = r(y)$

Input on result bus $[\![s, e_r, f_r]\!] =_{p-\hat{e}} x$
$f_r < 1 \Rightarrow e_r \leq e_{max}$
$f_r < 1 \leftrightarrow f_r[-1:0] = 00$

approximation of overflow:
$OVF1 = 1 \leftrightarrow 2^{e_r} f_r \geq 2^{e_{max}+1}$
$OVF2 = OVF \wedge \neg OVF1$
we use here OVF before rounding
$|\hat{r}(x)| > X_{max}$

Lemma 7.4 $OVF2 \rightarrow \hat{e} = e_{max} \wedge sigrd(s, \hat{f}) = 2$

Proof: $OVF(x) \leftrightarrow |\hat{r}(x)| > x_{max}$
$\hat{e} > e_{max} \vee \hat{e} = e_{max} \wedge sigrd(s, \hat{f}) = 2$
$\neg OVF1 \Rightarrow 2^{e_{max}+1} > 2^{e_r} f_r =_{p-\hat{e}} |x|$
$f_r \geq 1$ (otherwise $e_r < e_{max} \Rightarrow \neg OVF(x)$)
$e_r = e_{max}$
$f_r < 2$ (otherwise $OVF1(x)$)
$[\![x[n-1:0]]\!]_{bias} = \langle x \rangle - bias_n$

$bias = 2^{n-1} - 1$
$[\![x]\!]_{bias}[y]$
$\langle x \rangle = \langle y \rangle + \langle 1^{n-1} \rangle \bmod 2^n$
$$UNF = \begin{cases} TINY \wedge Loss & : \neg UNFen \\ TINY & : UNFen \end{cases}$$

x result $\quad \alpha = 3 * 2^{n-2} = \langle 110^{n-2} \rangle$
$$y = \begin{cases} 2^{-\alpha} * x & : OVF \wedge OVFen \\ 2^{\alpha} * x & : UNF \wedge UNFen \\ x & : otherwise \end{cases}$$

$\hat{\eta}(y) = (s, \hat{e}, \hat{f})$
$\eta(y) = (s, e, f)$
$$(p, n) = \begin{cases} (53, 11) & : db \\ (24, 8) & : \neg db \end{cases}$$

Input:
$[\![s, e_r, f_r]\!] =_{p-\hat{e}} x$
$fr[-1:0] = 00 \Rightarrow e_r \leq e_{max}$

$OVF1 \Leftrightarrow 2^{e_r} * f_r \geq 2^{e_{max}+1}$
$OVF2 \rightarrow \hat{e} = e_{max} \wedge sigrd(s, \hat{f}) = 2$
overflow after rounding

$$e_n = \begin{cases} e + a & : OVF2 \wedge OVFen \\ e & : otherwise \end{cases}$$

$f_n =_p f \qquad f_1 = [f_n]_p \qquad f_2 = sigrd(s, f_1)$
$(e_2, f_3) = post(e_n, f_2) \qquad SIGinx \Leftrightarrow f_1 \neq f_2$
$$(e_3, f_3) = \begin{cases} (e_{max} + 1 - \alpha, 1) & : OVF2 \wedge OVFen \\ (e_2, f_3) & : otherwise \end{cases}$$

rounder construction
$f_1 = [f]_p \qquad f_2 = sigrd(s, f)$
$$(e_2, f_3) = \begin{cases} post(e + a, sigrd(s, f)) & : OVF2 \wedge OVFen \\ post(e, sigrd(s, f)) & : otherwise \end{cases}$$
$(e_3, f_4) = post(e, sigrd(s, f))$
$(s, e_{out}, f_{out}) = (s, exprd(s, post(e, sigrd(s, f))))$

Normalisation Shifter
σ :left shift distance of SigNormShift
usually lz but: $TINY \wedge \overline{UNFen}$ →result denormal, $e = e_{min}$
$2^{e_r} f_r = 2^{e_{min}} * 2^{e_r - e_{min}} * f_r$
$e_r - e_{min} < 0$ possible right shift

$$y = \begin{cases} (-1)^s 2^{\hat{e}-a}\hat{f} & : OVF \wedge OVFen \\ (-1)^s 2^{\hat{e}-a}\hat{f} & : TINY \wedge UNFen \\ (-1)^s 2^{\hat{e}}\hat{f} & : otherwise \end{cases}$$

$$\hat{\eta}(y) = \begin{cases} (s, \hat{e}-\alpha, \hat{f}) & : OVF \wedge OVFen \\ (s, \hat{e}+\alpha, \hat{f}) & : TINY \wedge UNFen \\ (s, \hat{e}, \hat{f}) & : otherwise \end{cases}$$

y normal $\Rightarrow \hat{\eta}(y) = \eta(y)$
y denormal $\Rightarrow x = (-1)^s 2^{\hat{e}}\hat{f} = (-1)^s 2^{e_{min}} 2^{\hat{e}-e_{min}}\hat{f}$

$$\eta(y) = \begin{cases} (s, \hat{e}-\alpha, \hat{f}) & : OVF \wedge OVFen \\ (s, \hat{e}+\alpha, \hat{f}) & : TINY \wedge UNFen \\ (s, e_{min}, 2^{e_{min}-\hat{e}}\hat{f}) & : TINY \wedge \neg UNFen \\ (s, \hat{e}, \hat{f}) & : otherwise \end{cases}$$

$$e = \begin{cases} \hat{e}-\alpha & : OVF \wedge OVFen \\ \hat{e}+\alpha & : TINY \wedge UNFen \\ e_{min} & : TINY \wedge \neg UNFen \\ \hat{e} & : otherwise \end{cases}$$

$f' = f_r/2 \qquad f'[0:56] = f_r[-1:55]$
$lz = \#$ leading zeros $in f_r[-1:55]$
$\beta = e_r - lz + 1$
$2^{\hat{e}} * \hat{f} = |x| =_{p-\hat{e}} 2^{e_r} * f_r = 2^{e_r+1} f'$
$\Rightarrow \beta = \hat{e} \qquad 2^{lz} f' =_p \hat{f} = 2^{\beta} * 2^{lz} f'$

$$e = \begin{cases} \beta-\alpha & : OVF \wedge OVFen \\ \beta+\alpha & : TINY \wedge UNFen \\ e_{min} & : TINY \wedge \neg UNFen \\ \beta & : otherwise \end{cases}$$

$TINY \wedge \neg UNFen$:
$f = 2^{\hat{e}-e_{min}}\hat{f}$
$2^{\hat{e}}\hat{f} =_{p-\hat{e}} 2^{e_r} f_r$
$f = 2^{\hat{e}-e_{min}}\hat{f} =_{p-\hat{e}+e_{min}} 2^{e_r-e_{min}} * f_r$
$\hat{e} < e_{min} \Rightarrow e_{min} - \hat{e} > 0$
$f =_p 2^{e_r-e_{min}+1} f'$
$\sigma = e_r - e_{min} + 1$ if $TINY \wedge \neg OVFen$
$\langle f_n[0] - f_n[1:63] \rangle = f_n =_p f' 2^{\sigma}$
$OVF \leftrightarrow (e_r > e_{max}) \vee ((e_r = e_max) \wedge f_r[-1])$
$e_{max} = 2^{n-1} - 1 = \langle 1^{n-1} \rangle$
$[e_r[12:0]] > \langle 1^{n-1} \rangle \leftrightarrow \neg e_r[12] \wedge \bigvee_{i=n-1}^{12} e_r[i]$ input:
$(-1)^{s_r} 2^{e_r} f_r =_{p-\hat{e}} x$
$f_r[-1:0] = 00 \rightarrow e_r \leq e_{max}$

$$e_n \begin{cases} e-\alpha+1 & : OVF2 \wedge OVFen \\ e & : otherwise \end{cases}$$
$f_n =_p f$

$$\sigma = \begin{cases} e_r - e_{min} + 1 & : TINY \wedge \neg UNFen \\ lz & : otherwise \end{cases}$$

$\delta_n =_p f' 2^{\sigma} = f_e \qquad f' = f_r/2$

$TINY$ before rounding $\leftrightarrow \quad 2^{e_r} f_r \quad = \quad 2^{e_r+1} f' \quad < \quad 2^{e_{min}} \quad \leftrightarrow$
$\begin{cases} e_r + 1 < e_{min} & : f' \in [1,2) \\ e_r + 1 - lz < e_{min} & : f' \in [0,1) \end{cases}$
$\leftrightarrow e_r + 1 - lz - e_{min} < 0$
$e_{min} = 1 - bias \Rightarrow 1 - e_{min} = bias = \langle 1^{n-1} \rangle$

$bias - lz = \langle 1^{n-1} \rangle - \overline{[0lz[5:0]}] + 1$
$= \langle 10^{n-1} \rangle + [1^7 lz[5:0] + 1$
$= \begin{cases} \left[0^6 1 lz[5:0]\right] & : \neg db \\ \left[0^3 1^4 [5:0]\right] & : db \end{cases}$

$$\gamma = \begin{cases} -\alpha & : OVF \wedge OVFen \\ \alpha & : TINY \wedge UNFen \\ 0 & : otherwise \end{cases}$$

$[x]_{bias} = [y] \qquad \langle x \rangle = \langle y \rangle + bias \ mod \ 2^n = [x] \ mod \ 2^n$
$sum = e_r - lz + 1 + \gamma + bias \ mod \ 2^n$
$sum = e_r + 1 + 1[1\overline{lz[5:0]} + \gamma + bias + 1$
$\delta = \gamma + bias + 1$

$$\delta = \begin{cases} bias - \alpha + 1 & : OVF \wedge OVFen \\ bias + \alpha + 1 & : TINY \wedge UNFen \\ bias + 1 & : otherwise \end{cases}$$

$[sh] = -sh[12]2^{12} + \langle sh[11:0] \rangle = \langle sh[5:0] \rangle \ mod \ 64 = \sigma'$

Claim 8.10 $\quad f_s = \begin{cases} cls(f'\sigma') & : \sigma \geq 0 \\ clr(f', |\sigma| & : otherwise \end{cases}$

$crs(f'|\sigma|) = crs(f', -\sigma) = cls(f', \sigma \ mod \ 64) = cls(f', \langle sh[5:0] \rangle)$

$$\langle s_h \rangle = \begin{cases} \langle t \rangle & : \langle t \rangle \leq 63 \\ 63 & : otherwise \end{cases}$$
$$= \begin{cases} \sigma & : 0 \leq \sigma \leq 63 \\ |\sigma| - 1 & : -64 \leq \sigma \leq -1 \\ 63 & : otherwise \end{cases}$$

Claim 8.10 $\sigma \leq 56$ no mistake done in left shifts
$lz \leq 56 \vee \sigma = e_r - e_{min} + 1$ if $TINY \wedge \neg UNFen$
$e = e_{min} \qquad f_r \neq 0$
$x =_{p-\hat{e}} 2^{e_r} f_r \qquad \hat{e} < e_{min}$
$|x| =_{p-e_{min}} 2^{e_r} f_r$

assume $e_r - e_m in + 1 \geq 56$
$e_r \geq e_{min} + 55$
$\Rightarrow |x| \geq 2^{e_{min}+55} - 2^{-55} \geq 2^{e_{min}}$

Half decoder $h[63:0] = 0^{64-\langle sh' \rangle} 1^{\langle sh' \rangle}$

$$u[0:63] = \begin{cases} 0^{64-\sigma} & : \sigma \geq 0 \\ 1^{|\sigma|} 0^{64-|\sigma|} & : -63 \leq \sigma \leq -1 \\ 1^{64} & : \sigma \leq -64 \end{cases}$$

$$u[0:63] = \begin{cases} 1^{64-\sigma} 0^{64+\sigma} & : \sigma \geq 0 \\ 0^{|\sigma|} 1^{64} 0^{64-|\sigma|} & : -63 \leq \sigma \leq -1 \\ 0^{64} 1^{64} & : \sigma \leq -64 \end{cases}$$

$\sigma = [sh[12:0]]$
$\sigma' = \sigma \ mod \ 64 = \langle sh[5:0] \rangle$
$f' = f_r/2$
$f_n =_p f' * 2^{\sigma}$

$$u = \begin{cases} 0^{64-\sigma}\ 1^{\sigma} & : 0 \leq \sigma \qquad (\leq 56) \\ 1^{|\sigma|}\ 0^{64-|\sigma|} & : -63 \leq \sigma \leq -1 \\ 1^{64} & : \sigma \leq -64 \end{cases}$$

$$vw = \begin{cases} 1^{64-\sigma}\ 0^{64+\sigma} & : 0 \leq \sigma \\ 0^{|\sigma|}\ 1^{64}\ 0^{64-|\sigma|} & : -63 \leq \sigma \leq -1 \\ 0^{64}\ 1^{64} & : \sigma \leq -64 \end{cases}$$

$$f_n[-1:126] = \begin{cases} f_l\ 0^{64} & : 0 \leq \sigma \\ f_l\ 0^{64-\sigma} & : -63 \leq \sigma \leq -1 \\ 0^{64}\ cls(f', \sigma') & : \sigma \leq -64 \end{cases}$$

$f_n =_p f_l$
$f_l = [f_r]_p$

$$\left.\begin{array}{ll} f_l\ [p-1] & : \text{last bit} \\ [p] & : \text{round bit} \\ [p+1] & : \text{sticky bit} \\ s & : \text{sign bit} \end{array}\right\} \Longrightarrow \text{increment or trancate} f[0:p-1]$$

Postnormalization

$$(e_3, f_3) = \begin{cases} (e_n + 2, 1) & : f_2 = 2 \\ (e_n, f_2) & : \end{cases}$$

Exponent adjustment

$$(e_3, f_4) = \begin{cases} (e_{max} + 1 - \alpha, 1) & : OVF2 \ \wedge \ OVFen \\ (e_2, f_3) & : \end{cases}$$

$f_3 = f_4 \qquad (f_2 = 1 \ \leftarrow \ SigOVF)$

$\boxed{OVF2 \ \leftrightarrow \ SigOVF \ \wedge \ [\![e_2 = e_2[10:0]]\!]_{bias} = e_{max} + 1}$ Wrong (s 336)

$[\![x]\!]_{bias} = \langle x \rangle - bias$
$e_{max} = \langle 1^n \rangle - 1 - bias$

$$\begin{aligned} e_{max} + 1 - \alpha &= e_{max} + 1 - 3 * 2^{n-2} \\ &= \langle 1^n \rangle - \langle 110^{n-2} \rangle - bias \\ &= \langle 001^{n-2} \rangle - bias \\ &= [\![001^{n-2}]\!]_{bias} \\ &= [\![00db^3 1^6]\!]_{bias} \end{aligned}$$

Exprd
... hidden bit $f_3[0] = 0 \ \Rightarrow \ e_r = 0...0$