CRAY Y-MP FLOATING RECIPROCAL APPROXIMATION OVERVIEW

The Floating Reciprocal unit is used in conjunction with the Floating Multiply unit to complete a divide sequence, for example:

$$\frac{10}{2} = 5 \qquad \text{as} \qquad \frac{10}{1} \times \frac{1}{2} = 5.0$$

The Floating Reciprocal unit is used to find a value which is equivalent to 1/2 or .5. The Floating Reciprocal unit would not come up with a value .5 but, an approximation of .5 which would be .4999999999. The reciprocal value of 1/2 or .4999999999 may then be multiplied by 10 using the Floating Multiply unit to get the value of 5.0 or 4.999999999.

When using the results of the Floating Reciprocal functional unit not all of the bits in the coefficient are used. In fact, the unit will output 33 bits of the coefficient of which only 30 bits are accurate. Thirty bits of accuracy is obtained from doing two iterations of Newton's formula inside the reciprocal unit. If 48 bits of accuracy is required, or full precision, a third iteration of Newton's formula would need to be performed. The third iteration is performed using the Floating Multiply functional unit.

The hardware will run the coefficient through two iterations of Newton's formula while the exponent will get complemented and a value of + 2 added to the complement. The exponent and coefficient are then sent to the Scalar/Vector registers. The Floating Reciprocal unit performs two iterations of Newton's formula which are as seen in the hardware:

$$-A_1 = -2A_0 + A_0^2B \qquad \text{First iteration}$$

$$-A_2 = -2A_1 + A_1^2B \qquad \text{Second iteration}$$

Looking closely at the two formulas you will notice three multiplies which are:

a) A_0^2B = A_1 pyramid

b) $A_1 * A_1$ = A_1^2 pyramid

c) A_1^2B = A_2 pyramid

The A_1 pyramid is formed by $A_0^2 * B_1$. A_0^2 is gotten from a look-up table along with $-2A_0$ on the (RA) option, and B_1 is 24 bits or bits (2^{47} - 2^{24}) of the original coefficient. These two operands are multiplied together. Once the multiply is complete the pyramid is summed using 3-bit adds and finalized by using 2-bit adds. While the adds are taking place the $-2A_0$, which is also looked-up, is added into the summation in two places along with some forced ones and zeroes. When the summation has completed, the first iteration of Newton's formula is complete, which is the following:

$$-A_1 = -2A_0 + A_0^2B$$

Hardware Trng.
YM03/03A J.E.S.

The second and third multiply pyramids are to form the second iteration which is:

$$-A_2 = -2A_1 + A_1^2B$$

Looking at the second iteration, there are two multiplies that take place. The first multiply is A_1 * A_1 for A_1^2, and the second multiply is taking A_1^2 and multiplying it by B2. The multiply of A1 * A_1 takes 18 bits of A_1 and multiplies it to itself. When multiplying a number by itself, two things happen: for one, the pyramid is duplicated in two places; and second, a portion of the pyramid does not need to be multiplied. Because the pyramid is duplicated, only half of the pyramid needs to be multiplied, and when adding the pyramid the summation is multiplied by two to pick up the other half that was not produced. To perform a multiply of two the summation would need to be shifted to the left by one place, or instead, shift the portion which does not need to be multiplied to the right one place. This requires fewer bits to be shifted and gives the same results.

Once A_1^2 pyramid summation has completed, the next step would be to perfom the multiply of A_1^2 * B_2 and while adding the A_2 pyramid put $-2A_1$ into the summation to form -A2 = $-2A_1 + A_1^2B$.

The third and final multiply will take 36 bits of the A_1^2 pyramid and multiply it by 36 bits of B_2. The bits that makeup B_2 are coefficient bits (2^{47} - 2^{11}). While the summation of the A_2 pyramid is taking place, $-2A_1$ is put into the summation into sum bit positions (2^{-2} - 2^{-17}). When the pyramid has been completely summed, bits (2^{47} - 2^{15}) are sent to the result register along with the exponent. Of the 33 coefficient bits only 30 are considered accurate.

Underflow/Overflow

When the Floating Reciprocal unit finds that the exponent is greater than or equal to 60000_8, the Floating Reciprocal unit will flag an overflow condition, and force the exponent to 60000 and 2^{47} equal to zero. The exponent range check is performed prior to complementing and adding of a plus two, to the exponent on the (RN) options. When the exponent is less than 20000_8 the Floating Reciprocal unit will flag an underflow condition and force the exponent again to 60000_8 and 2^{47} equal to zero. Both underflow and overflow conditions would send a flag to the exchange package.

Reciprocal Approximation Iteration Formula

PROBLEM: Given a number B such that $.5 \le B < 1$, find a number A such that $\frac{1}{A} = B$.

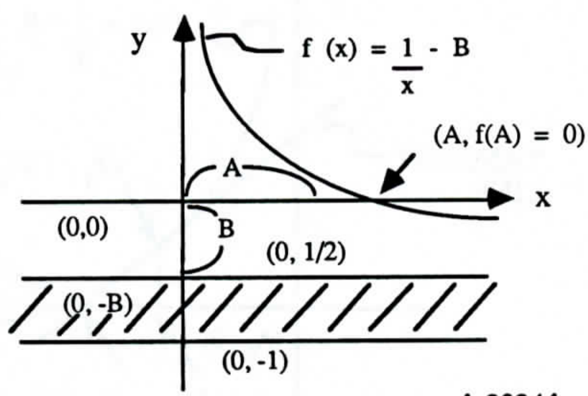
(Note that the restriction on the domain of B is only natural, since any normalized coefficient will always lie between .5 and 1.)

To find a good approximation value to A, first consider the function $f(x) = \frac{1}{x} - B$.

Now find a number A such that $f(A) = \frac{1}{x} - B = 0$. In other words, the number A we are seeking is the root of the equation $\frac{1}{x} - B = 0$.

From the graph, the value of A is the x-coordinate of the point located at the intersection of the curve of f (x) and the x-axis. To find the approximate value of A, use Newton's method for approximating the real roots of an arbitary equation $f(x) = 0$:

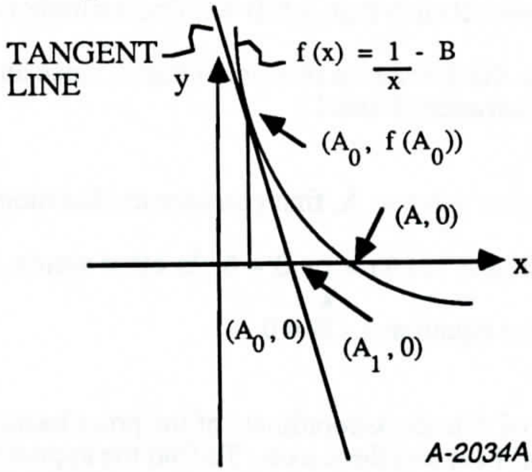
Step 1: Select an initial value A_0 (by using the look-up tables in CRAY Hardware) which is close to the value of A.



A-2034A

YM04/01

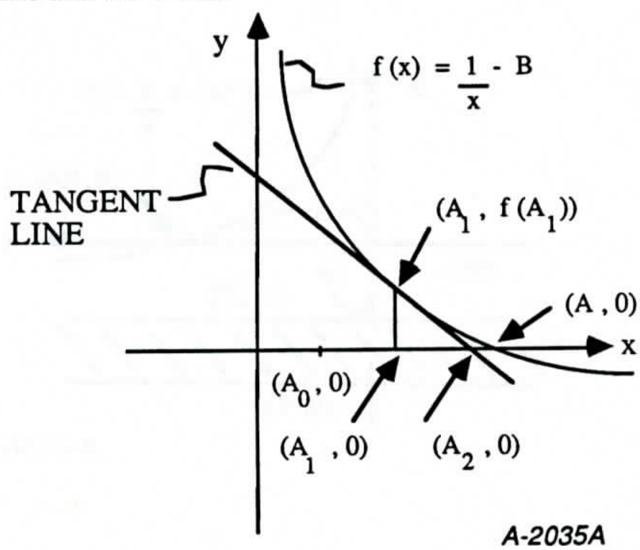
Step 2: Draw a tangent line at $(A_0,\ f(A_0))$ and find A_1, where $(A_1, 0)$ is the intersection of the tangent line and the x-axis.



From the equation of the tangent line $y = \frac{-x}{A_0^2} - B + \frac{2}{A_0}$, substituting $x = A_1$, $y = 0$,

and factoring out A_0^2 we find that $A_1 = 2A_0 - A_0^2 B$.

Step 3: Draw a tangent line at $(A_1,\ f(A_1))$ and find A_2, where $(A_2, 0)$ is the intersection of the tangent line and the x-axis.



From the equation of the tangent line $y = \frac{-x}{A_1^2} - B + \frac{2}{A_1}$, substituting $x = A_2$, $y = 0$,

and factoring out A_1^2, we find that $A_2 = 2A_1 - A_1^2 B$.

As you see, A_2 is much closer to A than either A_0 or A_1. By repeating this process, a good approximation to the value of A can be found.

YM04/02

NEWTON'S FORMULA DERIVATION

$A_{(i+1)} = 2A_i - A^2{}_iB,$

where:

$A_{(i+1)}$ = Next iteration — A_0 and A_0^2 in look-up table

A_i = Current iteration

B = Divisor

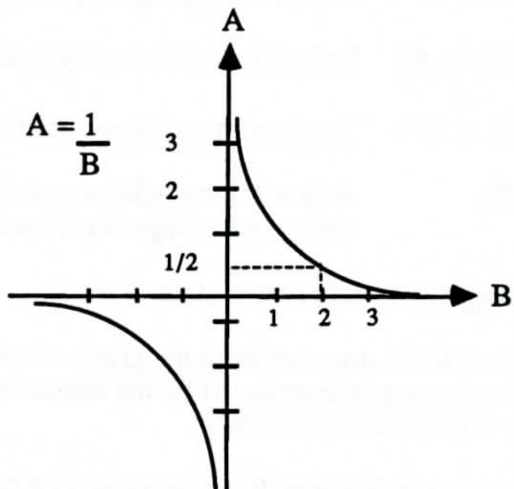
An initial guess starts the process with A_0, i.e.

A_1	$= 2A_0 - A_0^2 B$	1st approximation
A_2	$= 2A_1 - A_1^2 B$	2nd approximation result of reciprocal unit
A_3	$= 2A_2 - A_2^2 B$	3rd approximation
A_n	$= 2A_{(n-1)} - A^2{}_{(n-1)} B$	nth approximation

The more iterations, the closer the approximation approaches true reciprocal value; but the closer the initial guess A_0 is to the final result, the fewer the number of iterations needed to obtain a good approximation.

Further Explanation and Examples

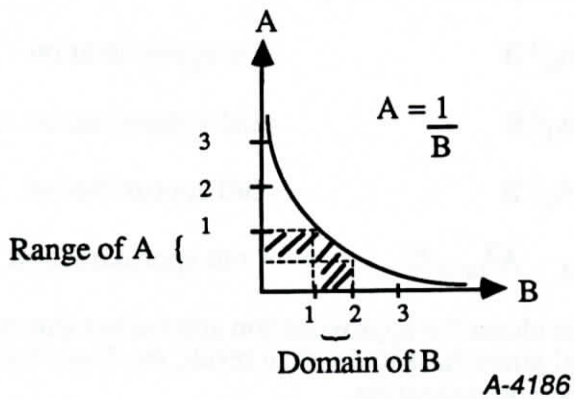
As can be seen from the foregoing discussion, the CRAY computes the reciprocal A of a number B by approximating the solution to the equation $\frac{1}{x} - B = 0$, using Newton's approximation method. This is the same as trying to read the value of A from the graph of the hyperbola $A = \frac{1}{B}$:



A-4185

YM04/03A

Since the CRAY uses only the absolute of B, we can restrict ourselves to the right-hand branch of the hyperbola above. Also, since the natural domain of the normalized coefficient of B is such that $.5 \leq B < 1$ (or $.4_8 \leq B < 1$), this will only produce A values such that $1 < A \leq 2$. We can further restrict the range of possible A values to $.4_8 < A \leq 1$ by expanding the domain of B to $1 \leq B < 2$. This can easily be done by shifting the normalized coefficient of B one place to the left, resulting in the change of Bs domain from $.4 \leq B < 1$ to $1 \leq B < 2$. We now have the whole story: to find the reciprocal A of any number B by reading it from the graph of the hyperbola $A = \frac{1}{B}$, first normalize the coefficient of B, then shift it one place to the left. This will force B to lie between 1 and 2 ($1 \leq B < 2$), and A to lie between .4 and 1 ($.4 < A \leq 1$), as indicated below:



Now find the value of B on the horizontal axis, follow it up to where it intersects the curve, and then to the left to find the reciprocal on the A axis. For example, to find the reciprocal of 24.4375_{10}, do the following steps:

24.4375_{10}

1)	30.3400_8	Convert to octal number
2)	$.6070 \times 2^5$	Convert to normalized floating point number
3)	1.4160×2^4	Left shift coefficient by 1 place
4)	$.5172 \times 2^{-4}$	Read reciprocal from graph
5)	$.0247_8$	Adjust binary point as specified by exponent (In this case, right shift coefficient by 4 places)
6)	$.0408_{10}$	Convert back to decimal

In practice, however, the CRAY does not store the graph of a hyperbola internally, and so step 4 of the above example is replaced by the following steps of Newton's approximation method as discussed previously:

4a) Find largest B value in hardware lookup table that is ≤ 1.416 (In this case, $B = 1.414$).

YM04/04

4b) Compute

1st iteration: $A_1 = 2\,A_0 - A_0^2B$ using table values

$= -(-2A_0 + A_0^2B)$ for $-2A_0$ and A_0^2

$= -(.534 + .334400 \times 1.414)$

$= .524052$

4c) 2nd iteration: $A_2 = -(-2A_1 + A_1^2B)$

$= -(.527654 + .3417076 \times 1.416)$

$= .517116$

Note that the reason the CRAY obtains such a high degree of accuracy after only two iterations is because of the closeness of the initial guess A_0 to the final result.

YM04/05

HANDLING OF B's EXPONENT

B = 40000 + E 1XXXXX XXXXXX XXXXXX

Exponent Coefficient

Value Of B $= 2^E * .1XXX\text{--------}X$ Normalized Floating-Point Number

$B = 2^{E-1} * 1.XXX\text{-------}X$ Left Shift Coefficient By 1

Let $b = 1.XXX\text{-------}X$ Rename Coefficient

Then $B = 2^{E-1} * b$

$$\frac{1}{B} = \frac{1}{2^{E-1} * b} = \frac{1}{2^{E-1}} * \frac{1}{b}$$

Let n = (E-1)

Simplify Negative Exponent

$$\frac{1}{2^n} = \frac{2^{-n}}{1} \quad \text{OR} \quad \frac{1}{2^{(E-1)}} = \frac{2^{-(E-1)}}{1} = \frac{2^{-E+1}}{1}$$

$$\frac{1}{B} = \frac{2^{-E+1}}{1} * \frac{1}{b}$$

Another method which is used in the mainframe: perform ones complement plus 2.

Example:	51132	Exponent
	26645	Peform ones Complement
	+ 2	Add One For Normalization Add One For twos Complement
	26647	

YM04/06B

CRAY Y-MP FLOATING RECIPROCAL FUNCTIONAL UNIT INSTRUCTIONS

Scalar

070ij0 Floating Reciprocal Approximation of (Sj) TO Si

174ij0 Floating Reciprocal Approximation of (Vj Elements) to Vi

CAL Format

Si /HSj Scalar

Vi /HVj Vector Reciprocal

YM04/07

CRAY Y-MP FLOATING RECIPROCAL OPTIONS USED

OPTION TYPE	NUMBER USED	DESCRIPTION
RA	3	A_0, A_0^2
RB	3	$A_0^2 * B_1$
RC	1	$A_0^2 * B_1$
RD	2	A_1 or A_1^2 Final Summation
RE	1	A_1^2 Summation
RF	1	A_1^2 Summation
RG	2	$A_1^2 * B_2$ First Level Sum
RH	3	$A_1^2 * B_2$ First Level Sum
RI	1	$A_1^2 * B_2$ First Level Sum
RJ	2	$A_1^2 * B_2$ First Level Sum
RK	1	$A_1^2 * B_2$ First Level Sum
RL	3	Final Coefficient Sum lower
RM	1	Final Coefficient Sum upper
RN	1	Result Exponent
RO	1	Vector Pop
	26 Total options	

YM04/08A

CRAY Y-MP FLOATING-POINT RECIPROCAL OPTIONS INVOLVED

(RA1, RA2, RA3) Options

The three (RA) options work in parallel and output the same results at the same time; however, the outputs go to different locations. The (RA1) sends its outputs to the (RB1) option, the (RA2) sends its results to the (RB2) options, and (RA3) sends its outputs to the (RB3) and (RC1) options. The (RA) chips receive the upper 7 bits of the coefficient, numbered (2^{46}-2^{40}), and from these bits the (RAs) will generate values for A_0^2 and A_0.

(RB1, RB2, RB3) Options

The (RB) options first fanout the B coefficient to form the A_1 pyramid and A_2 pyramid. The (RB1) option forms part of the A_1 pyramid which is generated from the logical products of A_0^2 and B_1. After the logical products are completed, the summation of the pyramid begins, using 3-bit adds. During the summation, $-2A_0$ is put into (RB1 and RB2) for rounding purposes, and the (RB3) adds forced ones into the summation. Because of the way the pyramid is formed, a portion of (RB3)'s output is not needed, and these terms are disabled by forcing zeroes into the appropriate inputs.

(RC1) Option

The (RC1) option handles the upper 9 bits of the A_1 pyramid. During the summation $-2A_0$ is added in to complete the first iteration $A_1 = -2A_0 + A_0^2B$. The (RC1) does not complete the summation on the upper end of the pyramid but sends its partial sums to the (RD1) option.

(RD1) Option

The (RD1) option completes the summation of the A_1 pyramid. The (RD1) receives its inputs from the (RB1, RB2, RB3) and (RC1) options. When the (RD1) has completed the summation for A_1, the (RD1) will fanout A_1 to the (RF1) and (RE1) to form the A_1^2 pyramid; also the (RD1) sends A_1 to the (RC1) option for a 5 CP delay, after which it is added into the A_2 pyramid. The (RD1) option also delays for 8 CPs, 6 bits of B_2, which are then sent to the A_2 pyramid.

YM04/09

(RD2) Option

The (RD2) option is identical to the (RD1) option except that the (RD2) option finalizes the upper half of the A_1^2 pyramid. There are forced zero and one conditions present to make the (RD2) option respond differently from the (RD1) option. After the (RD2) option has completed the summation of the A_1^2 pyramid, the (RD2) will fanout its results of A_1^2 to the (RG1), (RH1, RH2, RH3), (RI1), (RJ), (RK1) options to help form the A_2 pyramid. The A_2 pyramid is formed from the product of A_1^2 and B_2. The (RD2) option also delays for 8 CPs 6 bits of B_2, which are then sent to the A_2 pyramid.

(RE1) Option

The (RE1) option forms the lower half of the A_1^2 pyramid. The (RE1) finalizes the summation on bits (2^{-19}-2^{-36}) of A_1^2, which is then fanned out to the A_2 pyramid on the (RG2), (RH2), (RJ1, RJ2) and (RK1) options. The (RE1) also forces (2^{-34} and 2^{-35}) equal to zero, because whether or not the bits are set in the pyramid, the summation will always come up with zeroes in these positions.

(RF1) Option

The (RF1) option forms the upper half of the A_1^2 pyramid. The (RF1), however, does not finalize the summation of A_1 times A_1, or A_1^2, but sends its partial summation to the (RD2) for completion. The (RF1 and RE1) options form the A_1^2 pyramid. However, not all the bits within the pyramid are actually produced, since this would result in a considerable amount of duplication for two reasons. The commutativity of multiplication assures us that when both operands are the same, bit 2^1 * bit 2^2 will yield the same result as bit 2^2 * bit 2^1, and therefore the result of such a multiplication does not need to be computed twice. Also, the logical product of a bit with itself is the same bit (i.e., 1 * 1 = 1 and 0 * 0 = 0), and so all products of the form bit 2^N * bit 2^N do not need to be computed at all, since the answer is always bit 2^N.

(RG1, RG2) Options

The (RG) options form part of the A_2 pyramid, which is made from $A_1^2 * B_2$. Once the pyramid is formed, the (RG) option begins summing the pyramid. The (RG) options sum bit positions (2^{-34}-2^{-27}) of the A_2 pyramid. The (RG) options do not complete the summation but send their partial summation to the (RL1, RL2) options for completion.

YM04/10

(RH1, RH2, RH3) Options

The (RH) options are identical to the (RG) options except that the input terms that make-up (A0 - A12) and (B0 - B12) are complemented. Since many of the terms that would be produced by the (RH2) option are not used because of their location in the pyramid, these outputs are disabled by forcing ones into the appropriate input. The partial summation of A_2 is sent to the (RL1, RL2, RL3, RM1) options for completion.

(RI1) Option

The (RI1) option handles the upper 13 bits of the A_2 pyramid. The (RI1) computes the product of $A_1^2 * B_2$, and partially sums the pyramid. The partial sum is then sent to the (RM1 and RL3) options for completion.

(RJ1, RJ2) Option

The (RJ) options handle the least significant bits of the pyramid, which take care of rounding up sum bits (2^{-38} - 2^{-32}) of the A_2 pyramid, so that the final pyramid contains only bit positions (2^{-1} - 2^{-33}), though only bits (2^{-1} - 2^{-30}) are considered accurate. The (RJ) options do not complete the summation, but sends their partial sum to the (RL1) option for completion.

(RK1) Option

The (RK1) option handles three different areas of the A_2 pyramid. One of the positions is sum bits (2^{-20} - 2^{-13}), another is sum bit positions (2^{-33} - 2^{-27}), and finally sum bit positions (2^{-38} - 2^{-34}). The (RK1) option does not finalize these positions, but passes its partial summations to the (RL1, RL2, RL3) options.

(RL1, RL2, RL3) Options

The (RL) options finalize the A_2 pyramid summation of $A_1^2 * B_2$. The (RL1) will output result bits (2^{20} - 2^{15}) to the Scalar or Vector registers, while (RL2) will output result bits (2^{28} - 2^{21}). (RL3) will add into its summation bits (2^{-17} - 2^{-12}) of $-2A_1$, and when the summation is completed it will output result bits (2^{36} - 2^{29}) to the Scalar or Vector registers. The outputs from the (RL) options are forced to zeros unless the control term Go Reciprocal is present.

(RM1) Option

The (RM1) option handles the upper summation of the A_2 pyramid, bit positions (2^{-2}-2^{-14}). While the summation is taking place, the (RM1) option adds in bits (2^{-2}-2^{-14}) of $-2A_1$, which it received from the (RC1) option after a 5 CP delay. When the (RM1) option has completed its summation, it will output bits (2^{46}-2^{37}) to the Scalar or Vector registers. The (RM1) option also delays exponent bits (2^{63}-2^{60}) for 12 CPs, and then sends them to the (RN1) option.

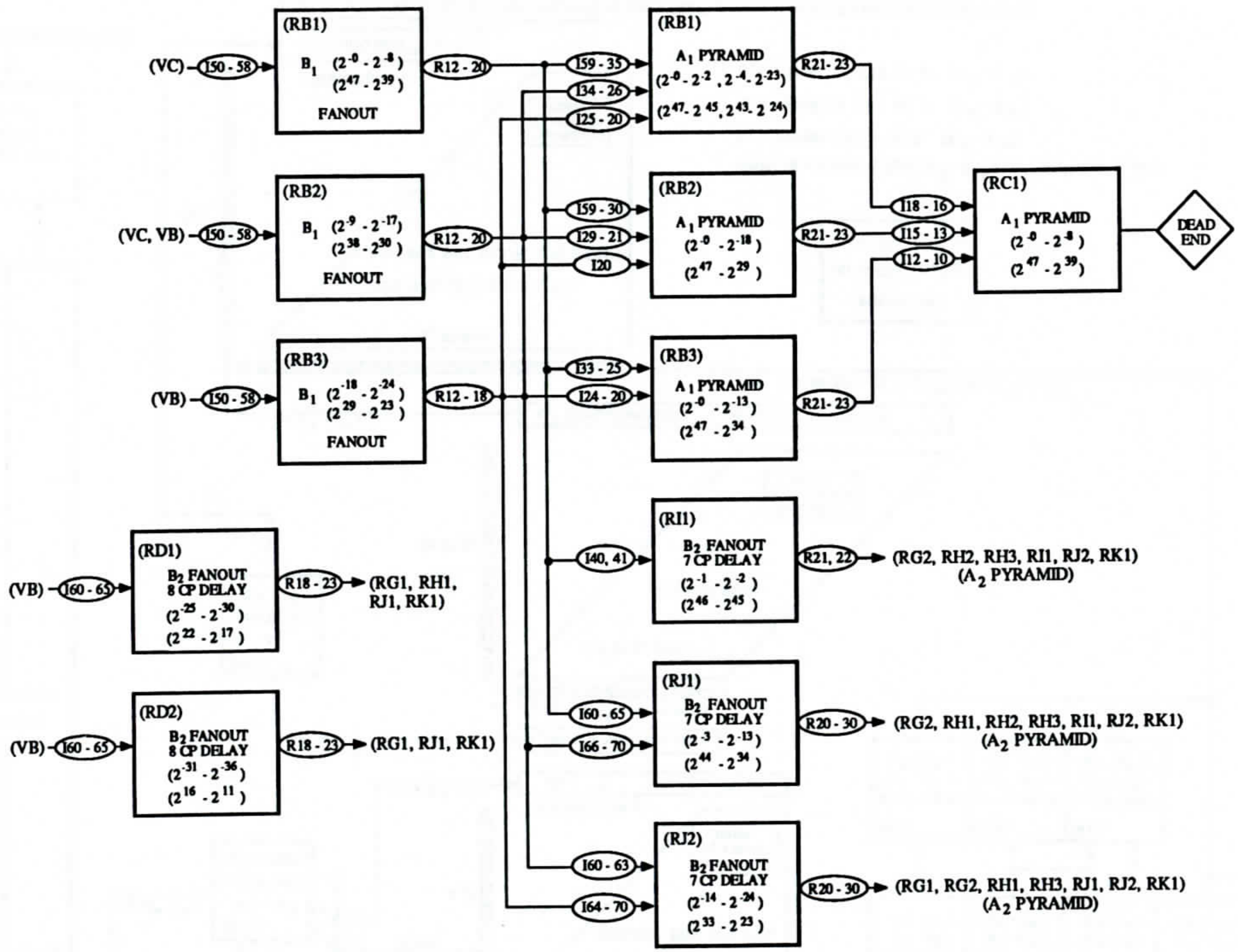
(RN1) Option

The (RN1) option delays the exponent bits (2^{59}-2^{48}) for 12 CPs, while the A_1, A_1^2 and A_2 pyramids are being formed and summed. Exponent bits (2^{63}-2^{60}) are delayed on the (RM1) option. After the 12 CP delay, the (RN1) option will receive all the bits of the Exponent (2^{63}-2^{48}), of which the sign bit (2^{63}) is left unchanged, and bits (2^{62}-2^{48}) are first complemented and then a value of (+2) is added into this complemented exponent. At CP 14, the exponent (bits 2^{62}-2^{48}), the Sign (bit 2^{63}), and bit (2^{47}), which is forced to a "1", are sent to the Scalar or Vector registers, provided the Go Reciprocal signal is present. Otherwise, the results are forced to zeroes. The (RN1) also performs a range check on the final exponent by checking bits (2^{62}, 2^{61}). If both of these bits are the same, a Range Error has occurred, in which event the resultant exponent is forced to 60000 and a Flag is sent to the Exchange package indicating a Floating-point Range Error.

(RO1) Option

The (RO1) option is the control for the Reciprocal Functional unit. The (RO1) fans out the Go Reciprocal control term, which allows the Reciprocal results to be sent to the Scalar or Vector registers. The (RO1) also decodes the k field of the instruction to determine whether to perform a Vector Pop Count or Floating Reciprocal operation. The (RO1) option contains the entire Vector Pop/Parity unit.

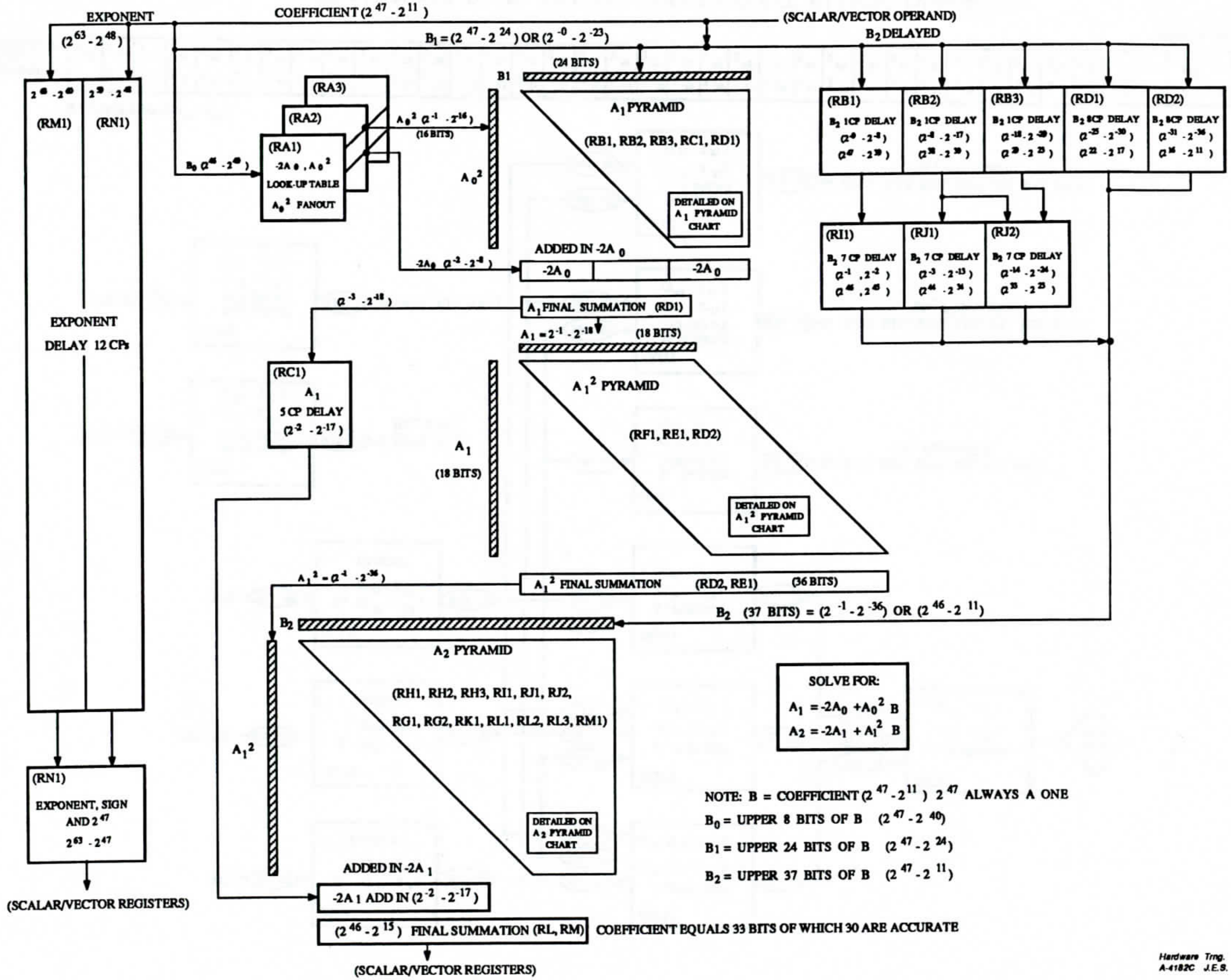
YM04/12A



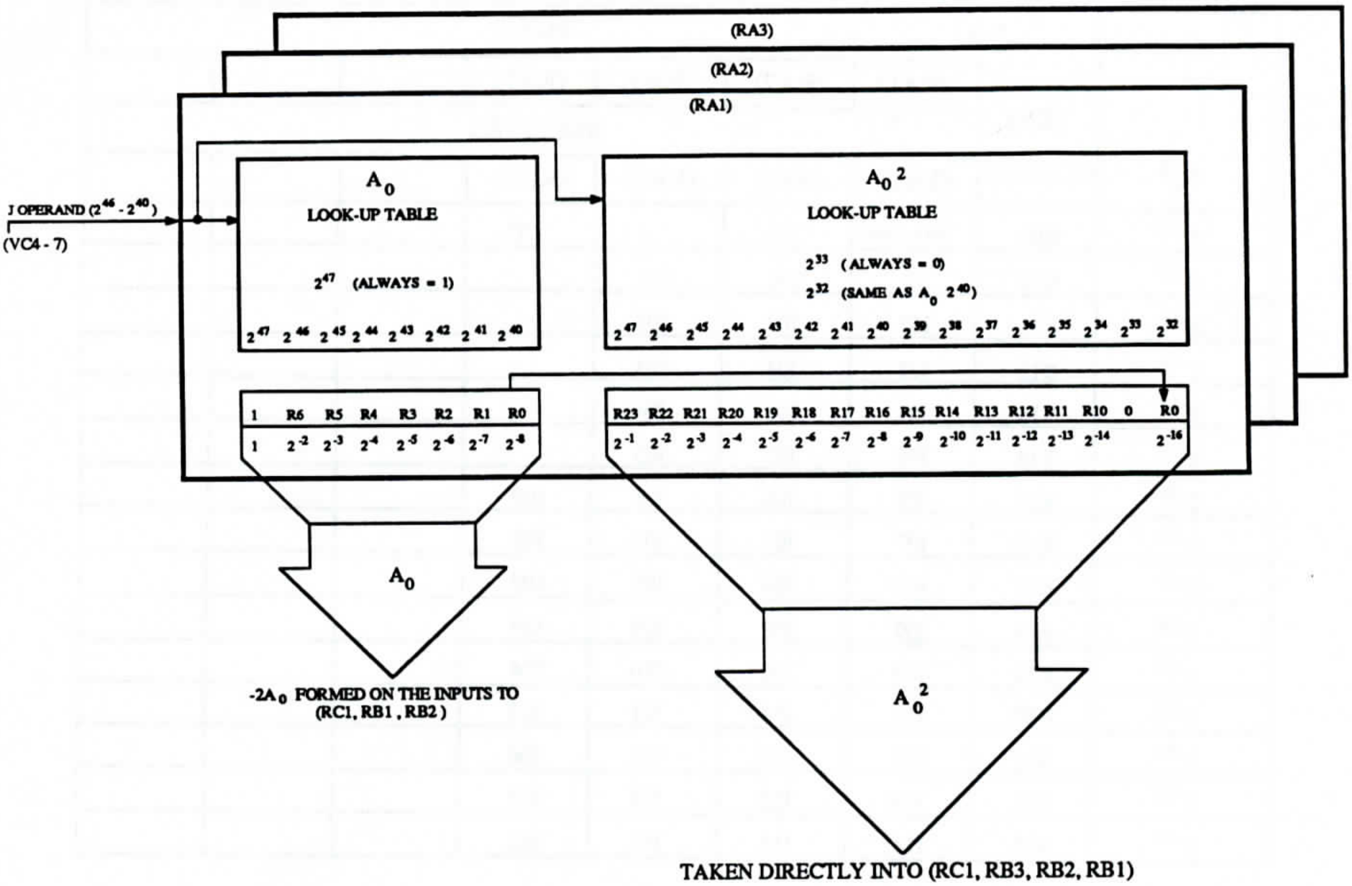
UNIT NUMBERING	2^{-0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}	2^{-22}	2^{-23}	2^{-24}	2^{-25}	2^{-26}	2^{-27}	2^{-28}	2^{-29}	2^{-30}	2^{-31}	2^{-32}	2^{-33}	2^{-34}	2^{-35}	2^{-36}
REGISTER NUMBERING	2^{47}	2^{46}	2^{45}	2^{44}	2^{43}	2^{42}	2^{41}	2^{40}	2^{39}	2^{38}	2^{37}	2^{36}	2^{35}	2^{34}	2^{33}	2^{32}	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}

Hardware Trng.
A-7436 D.E.

CRAY Y-MP B_1, B_2 DELAY AND FANOUT BLOCK DIAGRAM



CRAY Y-MP FLOATING RECIPROCAL UNIT FLOW CHART



Example of generating $A_0{}^2$ and -2A_0 : Sj or Vj = | 040000 | 152036 | 132461 | 000000 | (parcel format)

A) Take upper 8 bits of the coefficient = .1520 (in parcel format)

B) Shift 8 bits to the left by 1 = 1.520 (no longer in parcel format)

C) Look-up A_0 and $A_0{}^2$ for 1.520 = A_0 = .464 (A_0 only 8 bits)

$A_0{}^2$ = .271220 ($A_0{}^2$ only 16 bits)

D) A_0 becomes -2A_0 by complementing A_0 = (-A_0) and shifting A_0 to the left by 1 for a multiple of 2. = (-2A_0)

Example of .464:

1) .464 = .10011010

2) Complement = .01100101

3) Shifted by 1 = 0.1100101

4) 0.1100101 = .624 which is the value for -2A_0

5) .624 is numbered 0. 1 1 0 0 1 0 1 in the Boolean

2^{-1} 2^{-2} 2^{-3} 2^{-4} 2^{-5} 2^{-6} 2^{-7} 2^{-8}

E) $A_0{}^2$ is taken directly from (RA) 2^{33} (always = 0) and 2^{32} is the same as 2^{40} of A_0.

Hardware Trng.
A-4162B J.E.S.

CRAY Y-MP A_0 AND $A_0{}^2$ LOOK-UP TABLES

(RA) $A_0{}^2$ FANOUT

$A_0{}^2$	(RA) R TERMS	SENT						
		(RA1)	(RA2)	(RA3)	(RA3)			
		RECEIVED						
		(RB1)	(RB2)	(RB3)	(RC1)			
2^{-16}	R00	I00, I40	I00		I20			
2^{-14}	R10	I01	I01	I01				
2^{-13}	R11	I02	I02	I02				
2^{-12}	R12	I03	I03	I03				
2^{-11}	R13	I04	I04	I04				
2^{-10}	R14	I05	I05	I05				
2^{-09}	R15	I06	I06	I06	I00			
2^{-08}	R16	I07	I07	I07	I01			
2^{-07}	R17	I08	I08	I08	I02			
2^{-06}	R18	I09	I09	I09	I03			
2^{-05}	R19	I10	I10	I10	I04			
2^{-04}	R20	I11	I11	I11	I05			
2^{-03}	R21	I12	I12	I12	I06			
2^{-02}	R22	I13	I13	I13	I07			
2^{-01}	R23	I14	I14	I14	I08			

(RA) $-2A_0$ FANOUT

$-2A_0$ BITS	(RA) R TERMS	SENT						
		(RA1)	(RA2)	(RA3)				
		RECEIVED						
		(RB1)	(RB2)	(RC1)				
-8	R00	I00, I40	I00	I20				
-7	R01	I41		I21				
-6	R02	I42		I22				
-5	R03	I43		I23				
-4	R04	I44		I24				
-3	R05		I40	I25				
-2	R06		I41	I26				

Hardware Trng.
A-4166B J.E.S.

CRAY Y-MP FLOATING RECIPROCAL $A_0{}^2$ AND $-2A_0$ FANOUT

RECIPROCAL TABLE LOOK-UP VALUES

B	A_0	$A_0{}^2$	$-2A_0$
1.000	.776	.774004	.000
1.004	.772	.764044	.010
1.010	.766	.754144	.020
1.014	.762	.744304	.030
1.020	.756	.734504	.040
1.024	.752	.724744	.050
1.030	.750	.721100	.054
1.034	.744	.711420	.064
1.040	.740	.702000	.074
1.044	.734	.672420	.104
1.050	.732	.666644	.110
1.054	.726	.657344	.120
1.060	.722	.650104	.130
1.064	.720	.644400	.134
1.070	.714	.635220	.144
1.074	.710	.626100	.154
1.100	.706	.622444	.160
1.104	.702	.613404	.170
1.110	.700	.610000	.174
1.114	.674	.601020	.204
1.120	.672	.575444	.210
1.124	.666	.566544	.220
1.130	.664	.563220	.224
1.134	.660	.554400	.234
1.140	.656	.551104	.240
1.144	.652	.542344	.250
1.150	.650	.537100	.254
1.154	.646	.533644	.260
1.160	.642	.525204	.270
1.164	.640	.522000	.274
1.170	.636	.516604	.300
1.174	.632	.510244	.310
1.200	.630	.505100	.314
1.204	.626	.501744	.320
1.210	.624	.476620	.324
1.214	.620	.470400	.334
1.220	.616	.465304	.340
1.224	.614	.462220	.344
1.230	.612	.457144	.350
1.234	.610	.454100	.354
1.240	.604	.446020	.364
1.244	.602	.443004	.370
1.250	.600	.440000	.374
1.254	.576	.435004	.400

Reciprocal Table Look-Up Values (Continued)

B	A_0	$A_0 2$	$-2A_0$
1.260	.574	.432020	.404
1.264	.572	.427044	.410
1.270	.570	.424100	.414
1.274	.566	.421144	.420
1.300	.564	.416220	.424
1.304	.562	.413304	.430
1.310	.560	.410400	.434
1.314	.556	.405504	.440
1.320	.554	.402620	.444
1.324	.552	.377744	.450
1.330	.550	.375100	.454
1.334	.546	.372244	.460
1.340	.544	.367420	.464
1.344	.542	.364604	.470
1.350	.540	.362000	.474
1.354	.536	.357204	.500
1.360	.534	.354420	.504
1.364	.532	.351644	.510
1.370	.530	.347100	.514
1.374	.526	.344344	.520
1.400	.524	.341620	.524
1.404	.522	.337104	.530
1.410	.520	.334400	.534
1.414	.520	.334400	.534
1.420	.516	.331704	.540
1.424	.514	.327220	.544
1.430	.512	.324544	.550
1.434	.510	.322100	.554
1.440	.506	.317444	.560
1.444	.506	.317444	.560
1.450	.504	.315020	.564
1.454	.502	.312404	.570
1.460	.500	.310000	.574
1.464	.476	.305404	.600
1.470	.476	.305404	.600
1.474	.474	.303020	.604
1.500	.472	.300444	.610
1.504	.470	.276100	.614
1.510	.470	.276100	.614
1.514	.466	.273544	.620
1.520	.464	.271220	.624

Reciprocal Table Look-Up Values (Continued)

B	A_0	$A_0{}^2$	$-2A_0$
1.524	.462	.266704	.630
1.530	.462	.266704	.630
1.534	.460	.264400	.634
1.540	.456	.262104	.640
1.544	.456	.262104	.640
1.550	.454	.257620	.644
1.554	.452	.255344	.650
1.560	.452	.255344	.650
1.564	.450	.253100	.654
1.570	.446	.250644	.660
1.574	.446	.250644	.660
1.600	.444	.246420	.664
1.604	.442	.244204	.670
1.610	.442	.244204	.670
1.614	.440	.242000	.674
1.620	.436	.237604	.700
1.624	.436	.237604	.700
1.630	.434	.235420	.704
1.634	.434	.235420	.704
1.640	.432	.233244	.710
1.644	.430	.231100	.714
1.650	.430	.231100	.714
1.654	.426	.226744	.720
1.660	.426	.226744	.720
1.664	.424	.224620	.724
1.670	.422	.222504	.730
1.674	.422	.222504	.730
1.700	.420	.220400	.734
1.704	.420	.220400	.734
1.710	.416	.216304	.740
1.714	.416	.216304	.740
1.720	.414	.214220	.744
1.724	.412	.212144	.750
1.730	.412	.212144	.750
1.734	.410	.210100	.754
1.740	.410	.210100	.754
1.744	.406	.206044	.760
1.750	.406	.206044	.760
1.754	.404	.204020	.764
1.760	.404	.204020	.764
1.764	.402	.202004	.770
1.770	.402	.202004	.770
1.774	.400	.200000	.774

(RB) B_1 FANOUT SENT CP 2

		SENT						
		(RB1)	(RB2)	(RB3)				
		RECEIVED						
B_1 BITS	R TERMS	(RC1)	(RC1)	(RC1)				
2^{-0}	R21	I18						
2^{-1}	R22	I17						
2^{-2}	R23	I16						
2^{-3}	R21		I15					
2^{-4}	R22		I14					
2^{-5}	R23		I13					
2^{-6}	R21			I12				
2^{-7}	R22			I11				
2^{-8}	R23			I10				

(RB1) B_1 FANOUT SENT CP 1

		RECEIVED						
B_1, B_1 T5 BITS	R TERMS	(RB1)	(RB2)	(RB3)	(RJ1)	(RJ2)	(RI1)	
2^{-0}	R12	I59	I38	I33				
2^{-1}	R13	I60	I37	I32			I40	
2^{-2}	R14	I61	I36	I31			I41	
2^{-3}	R15		I35 I59	I30	I60			
2^{-4}	R16	I39	I34 I60	I29	I61			
2^{-5}	R17	I38	I33 I61	I28	I62			
2^{-6}	R18	I37	I32	I27 I59	I63			
2^{-7}	R19	I36	I31	I26 I60	I64			
2^{-8}	R20	I35	I30	I25 I61	I65			

(RB2) B_1 FANOUT SENT CP 1

		RECEIVED						
B_1 BITS	R TERMS	(RB1)	(RB2)	(RB3)	(RJ1)	(RJ2)		
2^{-9}	R12	I34	I29	I24	I66			
2^{-10}	R13	I33	I28	I23	I67			
2^{-11}	R14	I32	I27	I22	I68			
2^{-12}	R15	I31	I26	I21	I69			
2^{-13}	R16	I30	I25	I20	I70			
2^{-14}	R17	I29	I24			I60		
2^{-15}	R18	I28	I23			I61		
2^{-16}	R19	I27	I22			I62		
2^{-17}	R20	I26	I21			I63		

Hardware Trng.
A-4164B J.E.S.

CRAY Y-MP FLOATING RECIPROCAL B_1, B_2 FANOUT
(sheet 1 of 3)

(RB3) B_1 FANOUT SENT CP 1

B_1 BITS	R TERMS	RECEIVED						
		(RB1)	(RB2)	(RJ2)				
2^{-18}	R12	I25	I20	I64				
2^{-19}	R13	I24		I65				
2^{-20}	R14	I23		I66				
2^{-21}	R15	I22		I67				
2^{-22}	R16	I21		I68				
2^{-23}	R17	I20		I69				
2^{-24}	R18			I70				

(RD1) B_2 FANOUT SENT AFTER 8 CP DELAY

B_2 BITS	R TERMS	RECEIVED						
		(RG1)	(RH1)	(RJ1)	(RK1)			
2^{-25}	R18	I27	I20	I36				
2^{-26}	R19	I26		I35				
2^{-27}	R20	I25		I34	I39			
2^{-28}	R21	I24		I33	I38			
2^{-29}	R22	I23		I32	I37			
2^{-30}	R23	I22		I31	I36			

(RD2) B_2 FANOUT SENT AFTER 8 CP DELAY

B_2 BITS	R TERMS	RECEIVED						
		(RG1)	(RJ1)	(RK1)				
2^{-31}	R18	I21	I30	I35				
2^{-32}	R19	I20		I34				
2^{-33}	R20			I33				
2^{-34}	R21			I32				
2^{-35}	R22			I31				
2^{-36}	R23			I30				

Hardware Trng.
A-4164B J.E.S.

CRAY Y-MP FLOATING RECIPROCAL B_1, B_2 FANOUT
(sheet 2 of 3)

(RI1) B_2 FANOUT SENT AFTER 7 CP DELAY

B_2 BITS	(RI1) R TERMS	RECEIVED						
		(RH3)	(RH2)	(RI1)	(RK1)	(RG2)	(RJ2)	
2^{-1}	R21	I37	I31	I30	I56	I38	I44	
2^{-2}	R22	I36	I30	I29	I55	I37	I43	

(RJ1) B_2 FANOUT SENT AFTER 7 CP DELAY

B_2 BITS	(RJ1) R TERMS	RECEIVED						
		(RH3)	(RH2)	(RI1)	(RK1)	(RG2)	(RJ2)	(RH1)
2^{-3}	R20	I35	I29	I28	I54	I36	I42	
2^{-4}	R21	I34	I28	I27	I53	I35	I41	
2^{-5}	R22	I33	I27	I26	I52	I34	I40	
2^{-6}	R23	I32	I26	I25	I51	I33	I39	
2^{-7}	R24	I31	I25	I24	I50	I32	I38	I38
2^{-8}	R25	I30	I24	I23		I31	I37	I37
2^{-9}	R26	I29	I23	I22		I30	I36	I36
2^{-10}	R27	I28	I22	I21		I29	I35	I35
2^{-11}	R28	I27	I21	I20		I28	I34 I56	I34
2^{-12}	R29	I26	I20			I27	I33 I55	I33
2^{-13}	R30	I25			I42	I26	I32 I54	I32

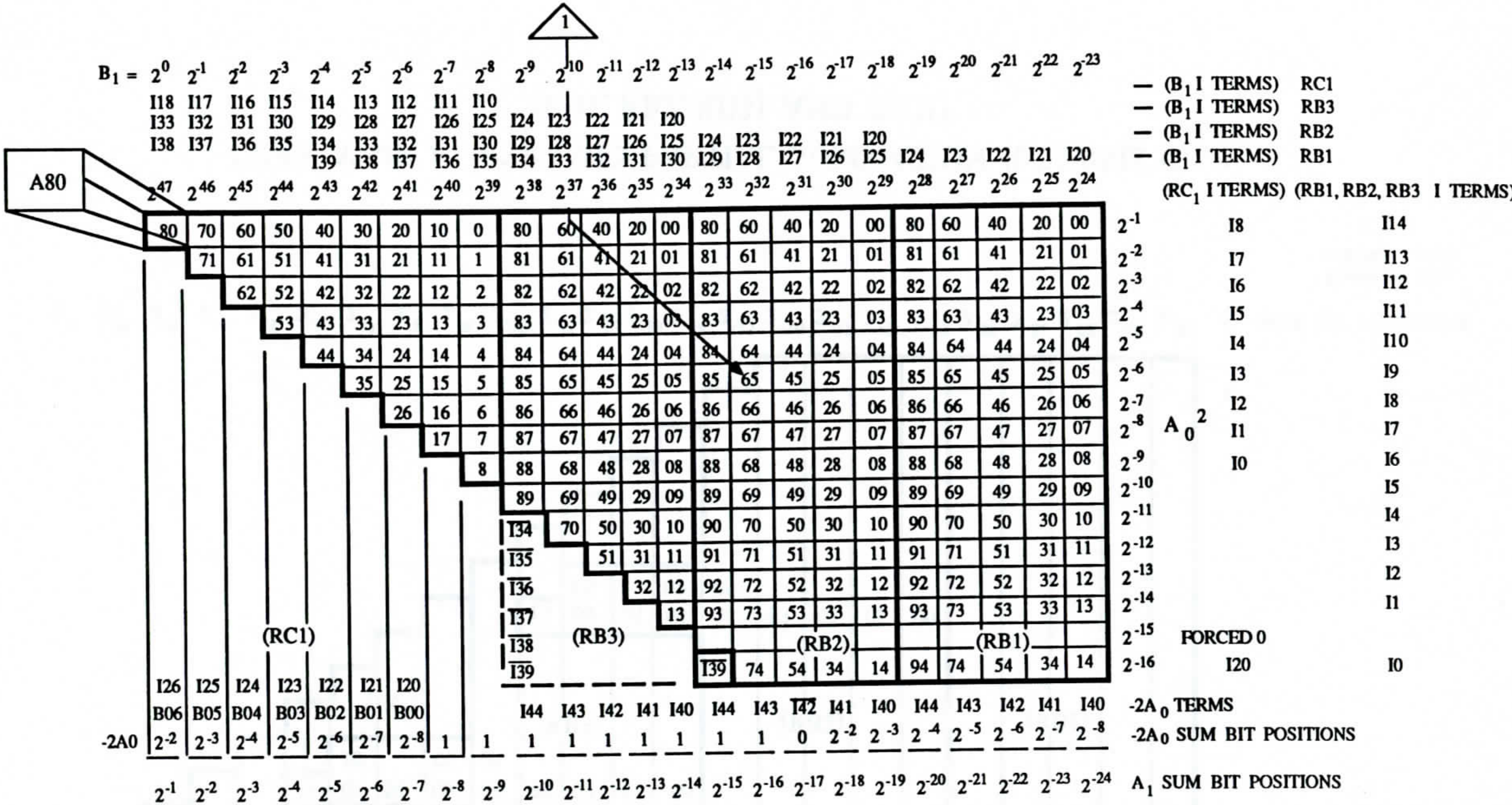
(RJ2) B_2 FANOUT SENT AFTER 7 CP DELAY

B_2 BITS	(RJ2) R TERMS	RECEIVED						
		(RH3)	(RH1)	(RJ2)	(RJ1)	(RG2)	(RG1)	(RK1)
2^{-14}	R20	I24	I31	I31 I53	I56	I25	I38	I41
2^{-15}	R21	I23	I30	I30 I52	I55	I24	I37	I40
2^{-16}	R22	I22	I29		I54 I45	I23	I36	
2^{-17}	R23	I21	I28		I53 I44	I22	I35	
2^{-18}	R23	I20	I27		I52 I43	I21	I34	
2^{-19}	R25		I26		I51 I42	I20	I33	
2^{-20}	R26		I25		I50 I41		I32	
2^{-21}	R27		I24		I40		I31	
2^{-22}	R28		I23		I39		I30	
2^{-23}	R29		I22		I38		I29	
2^{-24}	R30		I21		I37		I28	

Hardware Trng.
A-4164B J.E.S.

CRAY Y-MP FLOATING RECIPROCAL B_1 , B_2 FANOUT
(sheet 3 of 3)

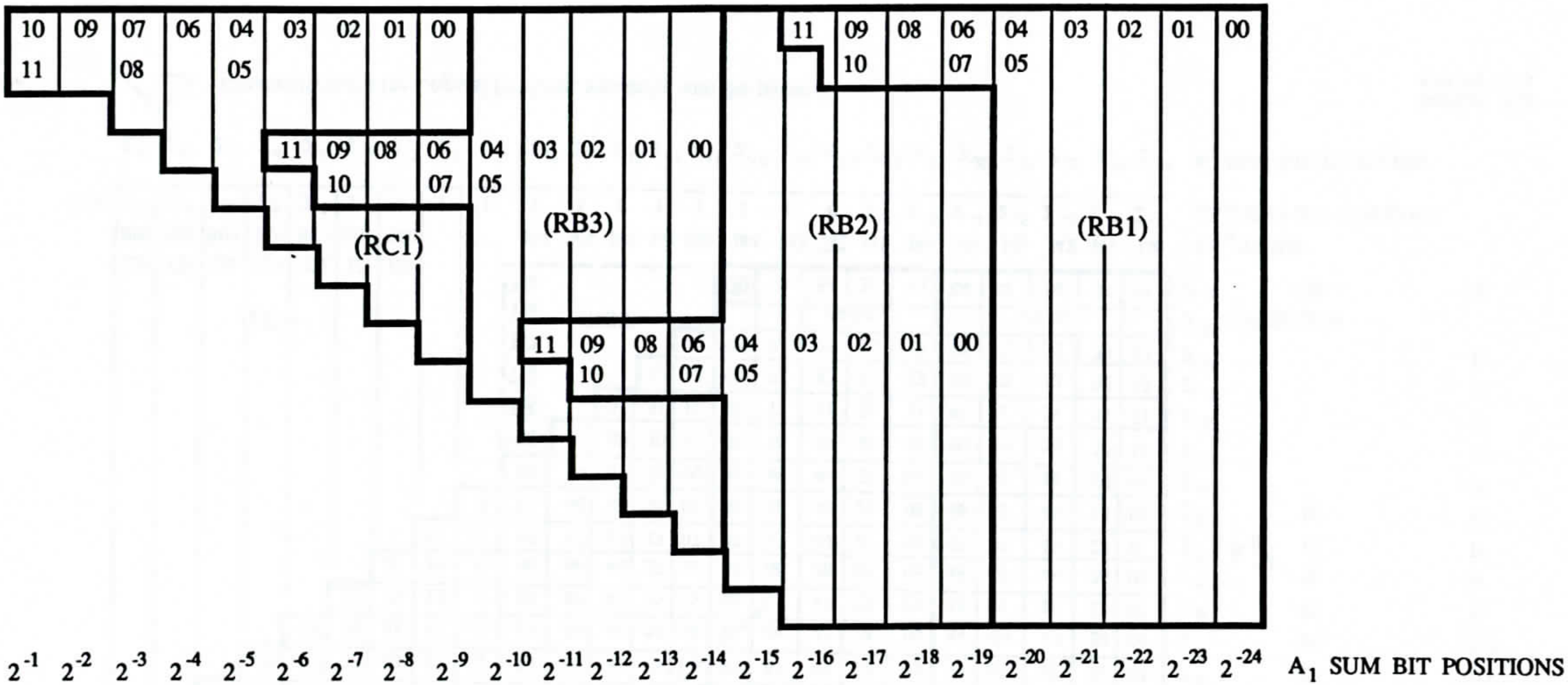
HTV-0834



1 Diagonally form the Logical Products, vertically sum the pyramid.

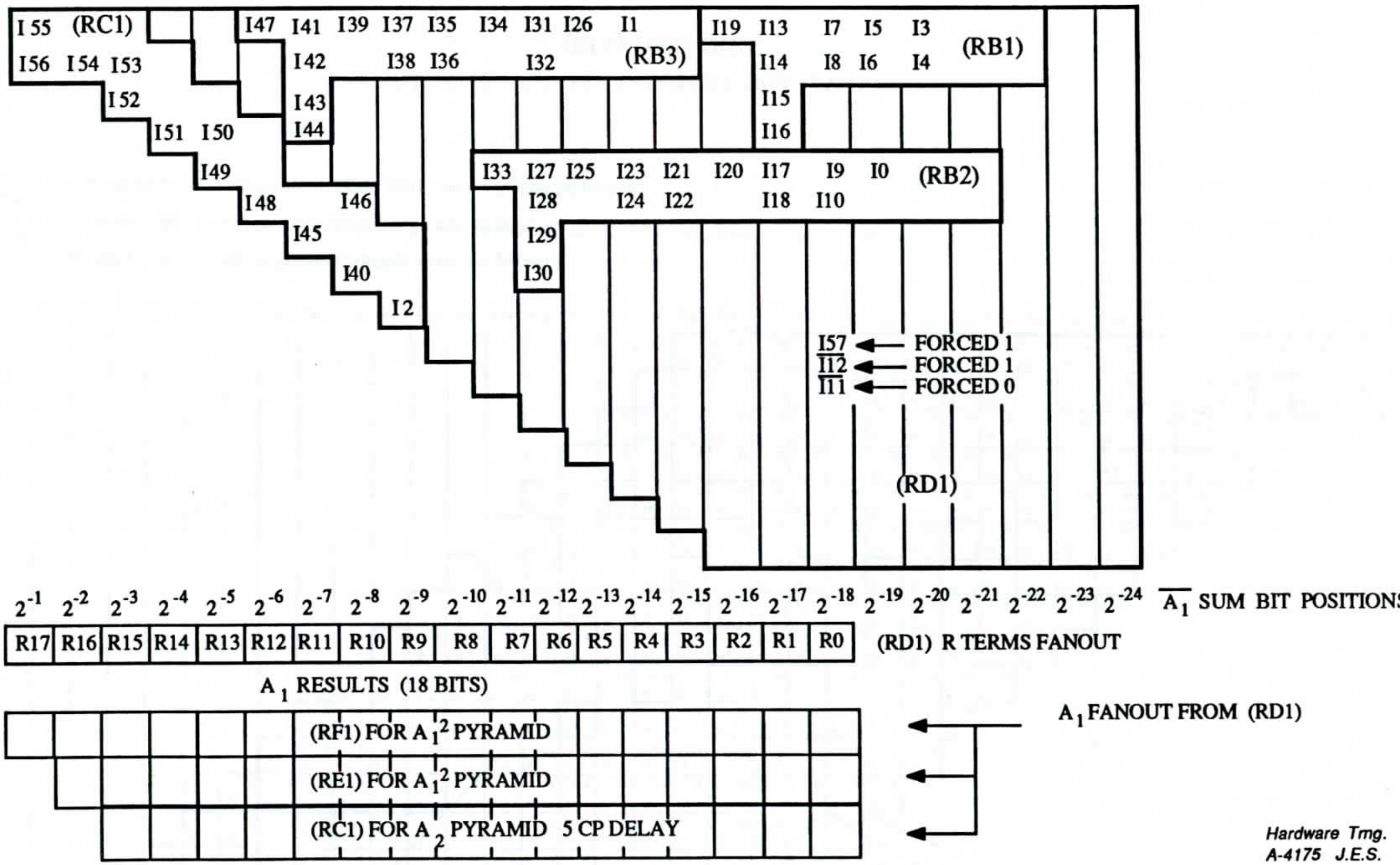
Hardware Trng.
A-4173D J.E.S.

CRAY Y-MP A_1 PYRAMID LOGICAL PRODUCTS (RB1, RB2, RB3)
AND (RC1) CP 1, CP 2

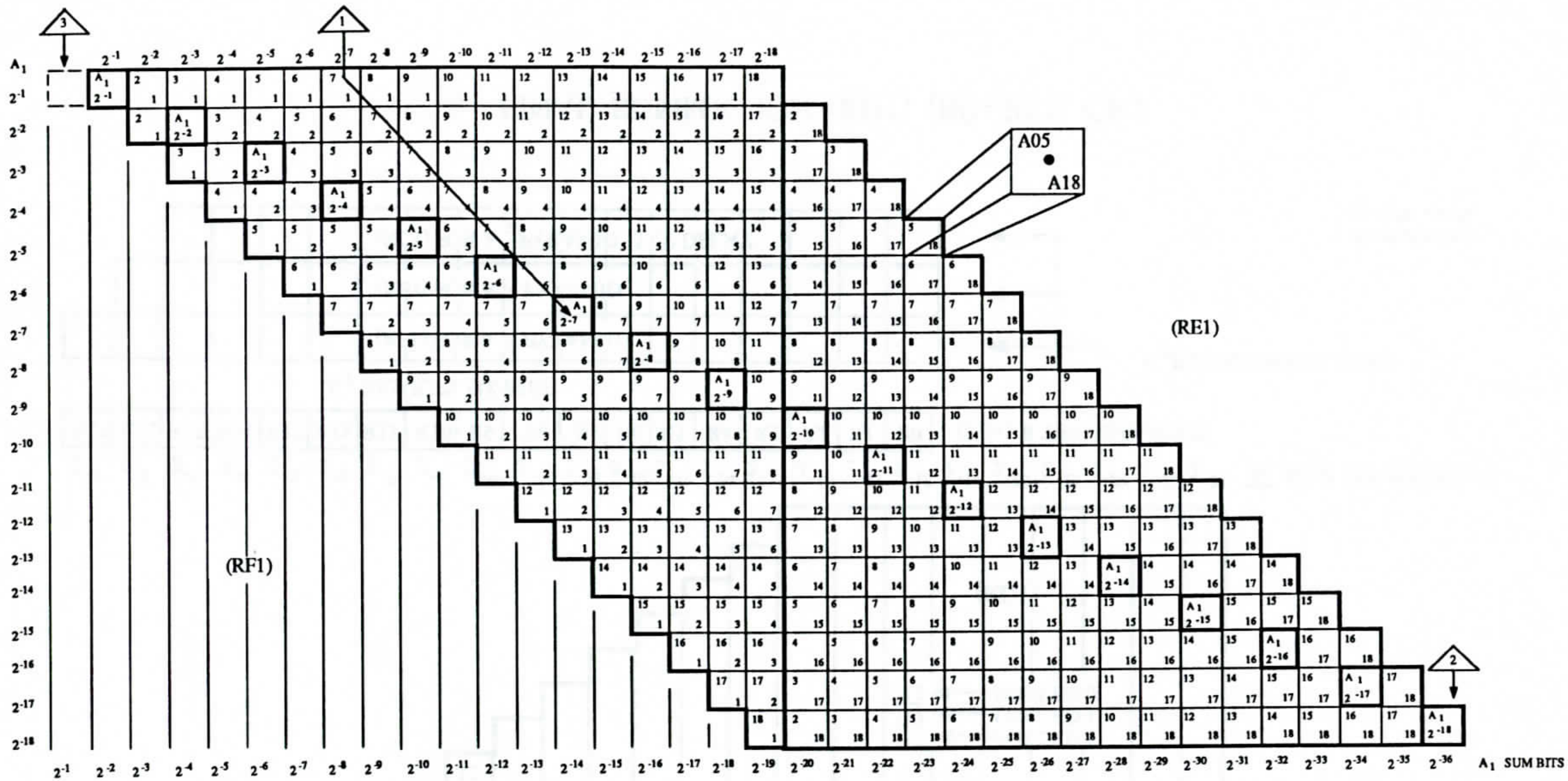


Hardware Trng.
A-4174A J.E.S.

CRAY Y-MP A_1 PYRAMID PARTIAL SUMMATION R TERMS CP 4
(RB1, RB2, RB3) AND (RC1)



A_1 FINAL SUMMATION (RD1) (R0 - R17) CP 5

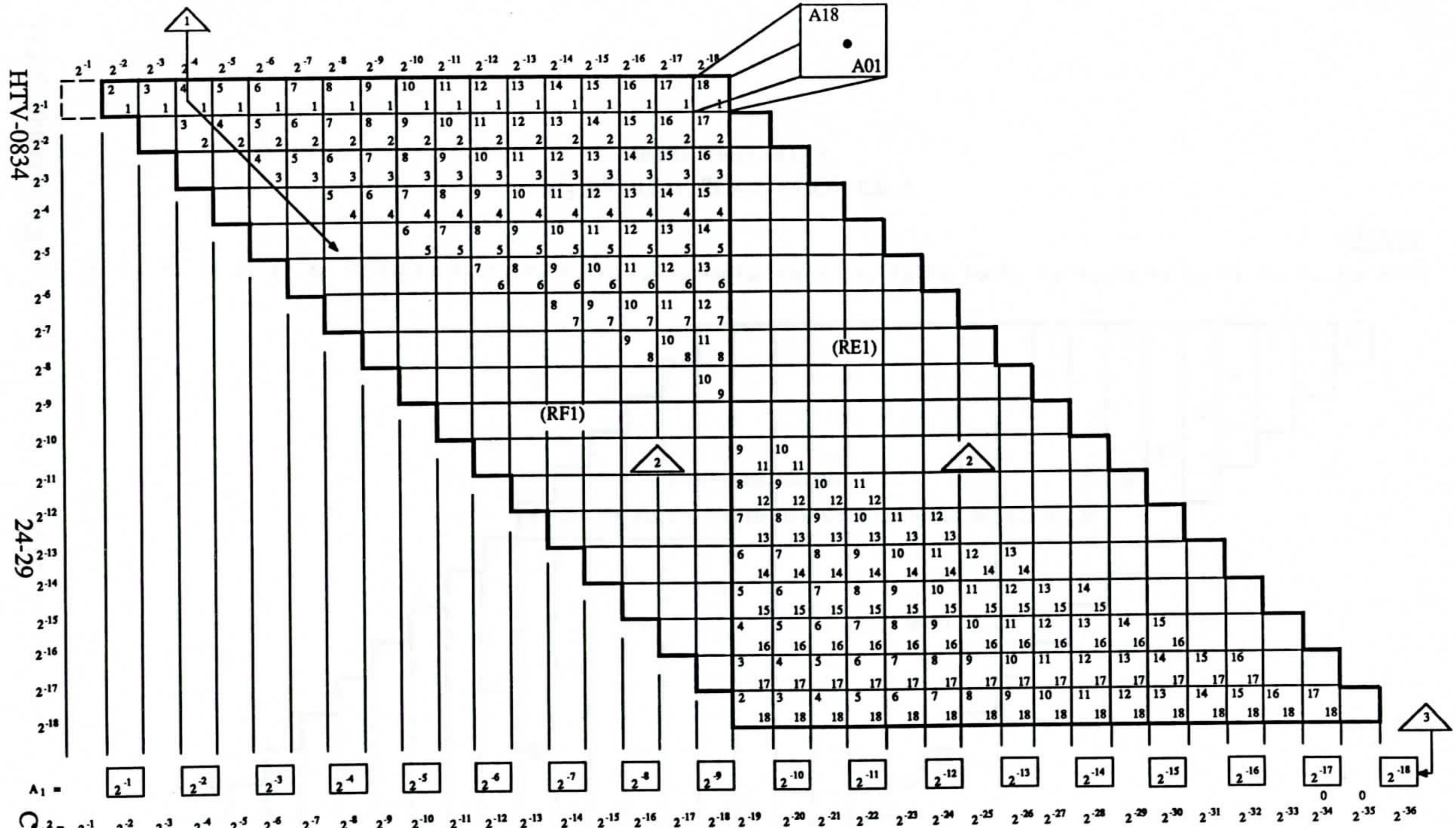


1. Diagonally form the Logical Products, vertically sum the pyramid.
2. The square of A_1 causes the same operand to be multiplied by itself. Example: Bit $2^{-18} * 2^{-18} = 2^{-18}$
3. Use all 36 units of the results including the carry into the highest bit position.

CRAY Y-MP A_1^2 FULL SUMMATION PYRAMID CP 6
(RF1) AND (RE1)

HTV-0834

24-29



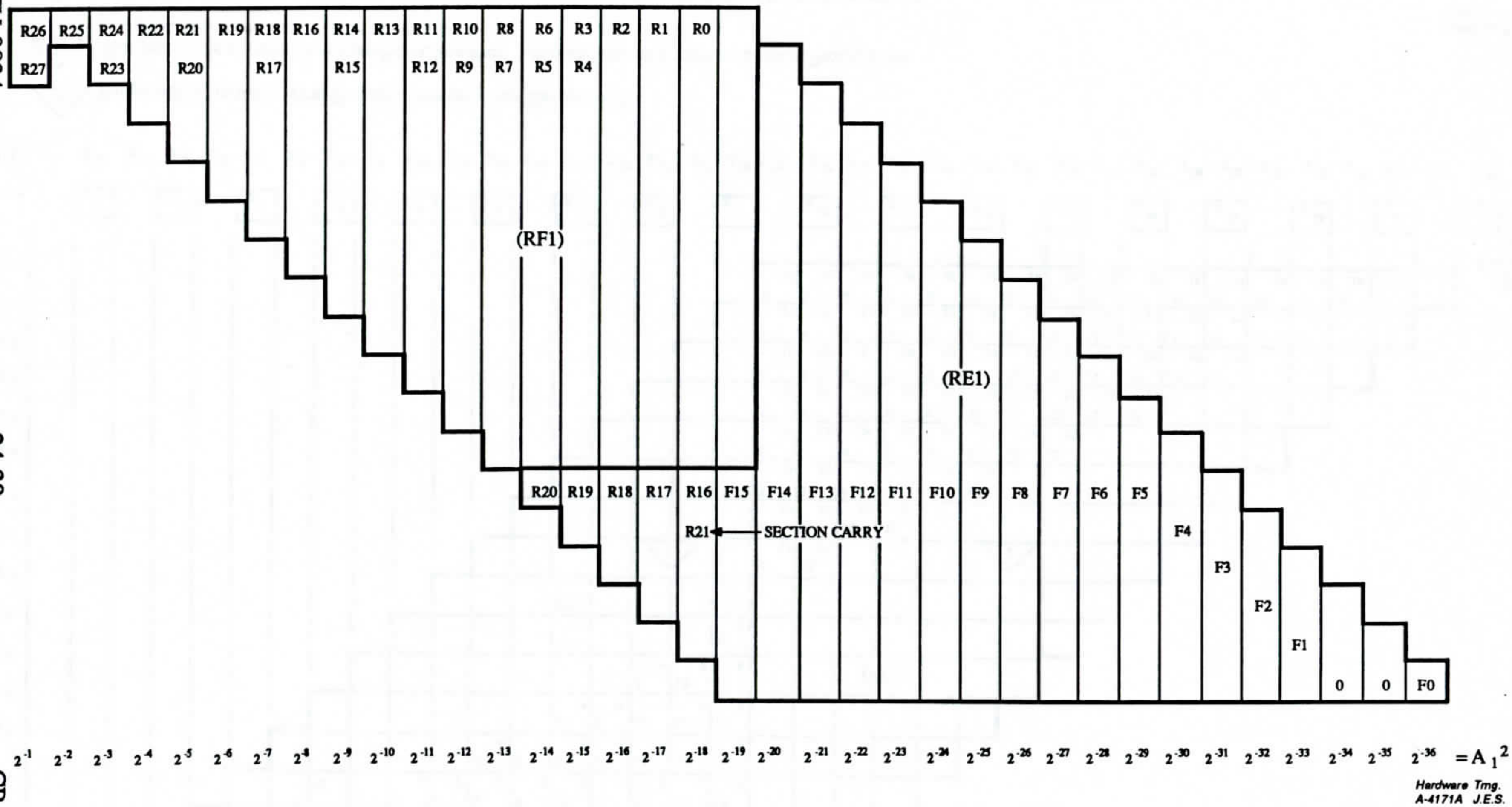
1 Diagonally form the Logical Products, vertically sum the pyramid.

2 This part of the pyramid is not produced, however, when any one of the shown Logical Products are produced, they have a sum value of two.

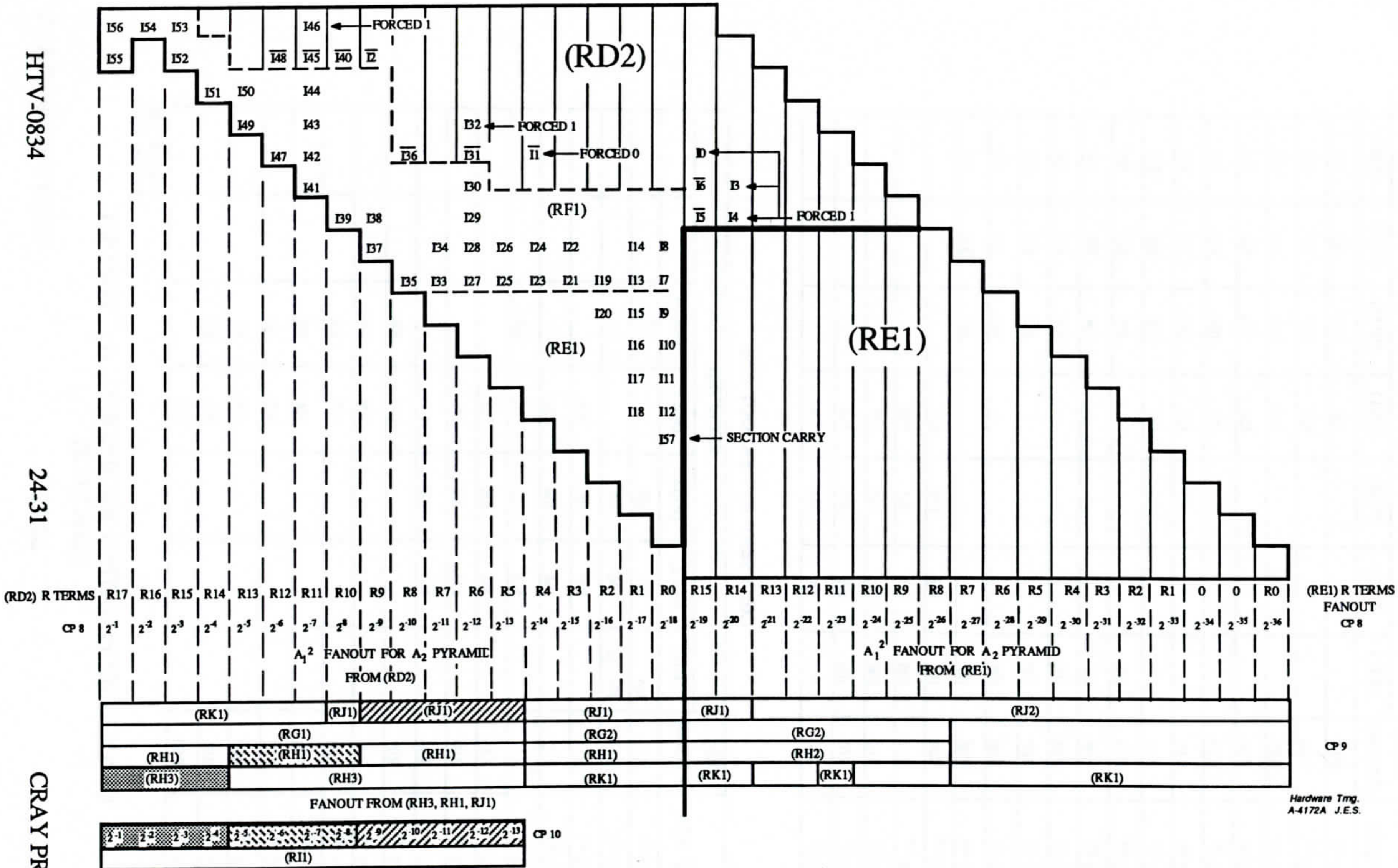
3 A_1 is added into the summation right shifted one place, the summation adjusted to give the same results as a Full Summation.

Hardware Trng.
A-4170A J.E.S.

CRAY Y-MP A_1^2 ESSENTIAL SUMMATION PYRAMID CP 6 (RF1) AND (RE1)



A_1^2 PARTIAL SUMMATION CP 7
(RF1) AND (RE1)



A_1^2 FINAL SUMMATION CP 8 AND
A_1^2 FANOUT TO FORM A_2 PYRAMID
CP 9 (RD2) AND (RE1)

(RD2) SENT A_1^2 FANOUT CP 8

A_1^2 BITS	(RD2) R TERMS	RECEIVED						
		(RJ1)	(RG2)	(RH2)	(RK1)	(RH3)	(RG1)	(RH1)
2^{-01}	R17				I26	I12	I12	I12
2^{-02}	R16				I25	I11	I11	I11
2^{-03}	R15				I24	I10	I10	I10
2^{-04}	R14				I23	I09	I09	I09
2^{-05}	R13				I22	I08	I08	I08
2^{-06}	R12				I21	I07	I07	I07
2^{-07}	R11				I20	I06	I06	I06
2^{-08}	R10	I12				I05	I05	I05
2^{-09}	R09	I11				I04	I04	I04
2^{-10}	R08	I10				I03	I03	I03
2^{-11}	R07	I09				I02	I02	I02
2^{-12}	R06	I08				I01	I01	I01
2^{-13}	R05	I07				I00	I00	I00
2^{-14}	R04	I06	I12	I12	I17			
2^{-15}	R03	I05	I11	I11	I16			
2^{-16}	R02	I04	I10	I10	I15			
2^{-17}	R01	I03	I09	I09	I14			
2^{-18}	R00	I02	I08	I08	I13			

(RE1) SENT A_1^2 FANOUT CP 8

A_1^2 BITS	(RE1) R TERMS	RECEIVED						
		(RJ1)	(RH2)	(RG2)	(RJ2)	(RK1)		
2^{-19}	R15	I01	I07	I07		I12		
2^{-20}	R14	I00 I20	I06	I06		I11		
2^{-21}	R13		I05	I05	I02			
2^{-22}	R12		I04	I04	I01			
2^{-23}	R11		I03	I03	I00	I10		
2^{-24}	R10		I02	I02	I12			
2^{-25}	R09		I01	I01	I11			
2^{-26}	R08		I00	I00	I10			
2^{-27}	R07				I09	I06		
2^{-28}	R06				I08	I05		
2^{-29}	R05				I07	I04		
2^{-30}	R04				I06	I03		
2^{-31}	R03				I05	I02		
2^{-32}	R02				I04	I01		
2^{-33}	R01				I03	I00		
2^{-36}	R00				I20			

Hardware Trng.
A-4165A J.E.S.

CRAY Y-MP FLOATING RECIPROCAL A_1^2 FANOUT
(sheet 1 of 2)

(RH3) SENT A_1^2 FANOUT CP 9

A_1^2 BITS	(RH3) R TERMS	RECEIVED (RI1)						
2^{-1}	R24	I12						
2^{-2}	R23	I11						
2^{-3}	R22	I10						
2^{-4}	R21	I9						

(RH1) SENT A_1^2 FANOUT CP 9

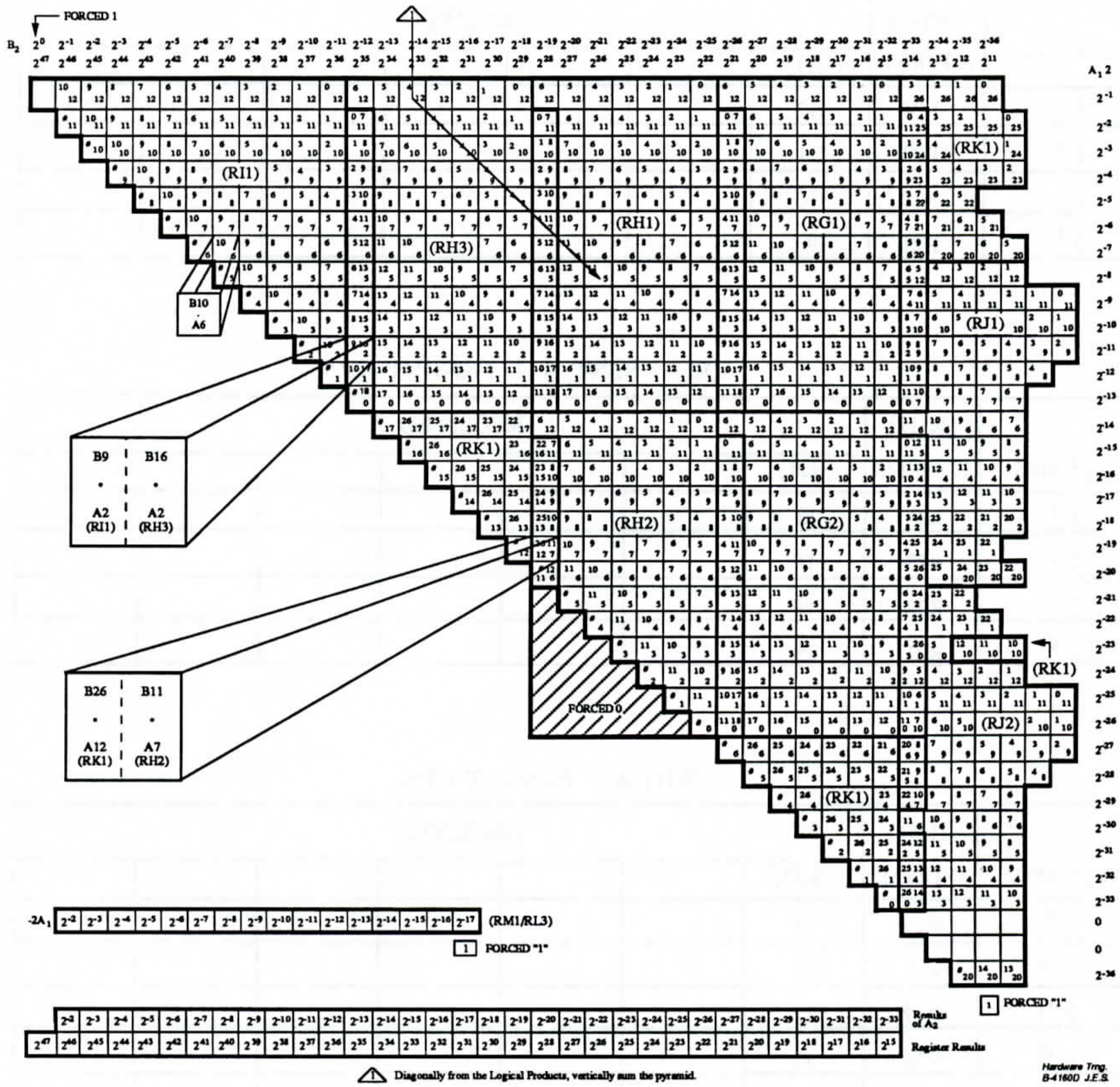
A_1^2 BITS	(RH1) R TERMS	RECEIVED (RI1)						
2^{-5}	R24	I8						
2^{-6}	R23	I7						
2^{-7}	R22	I6						
2^{-8}	R21	I5						

(RJ1) A_1^2 FANOUT CP 9

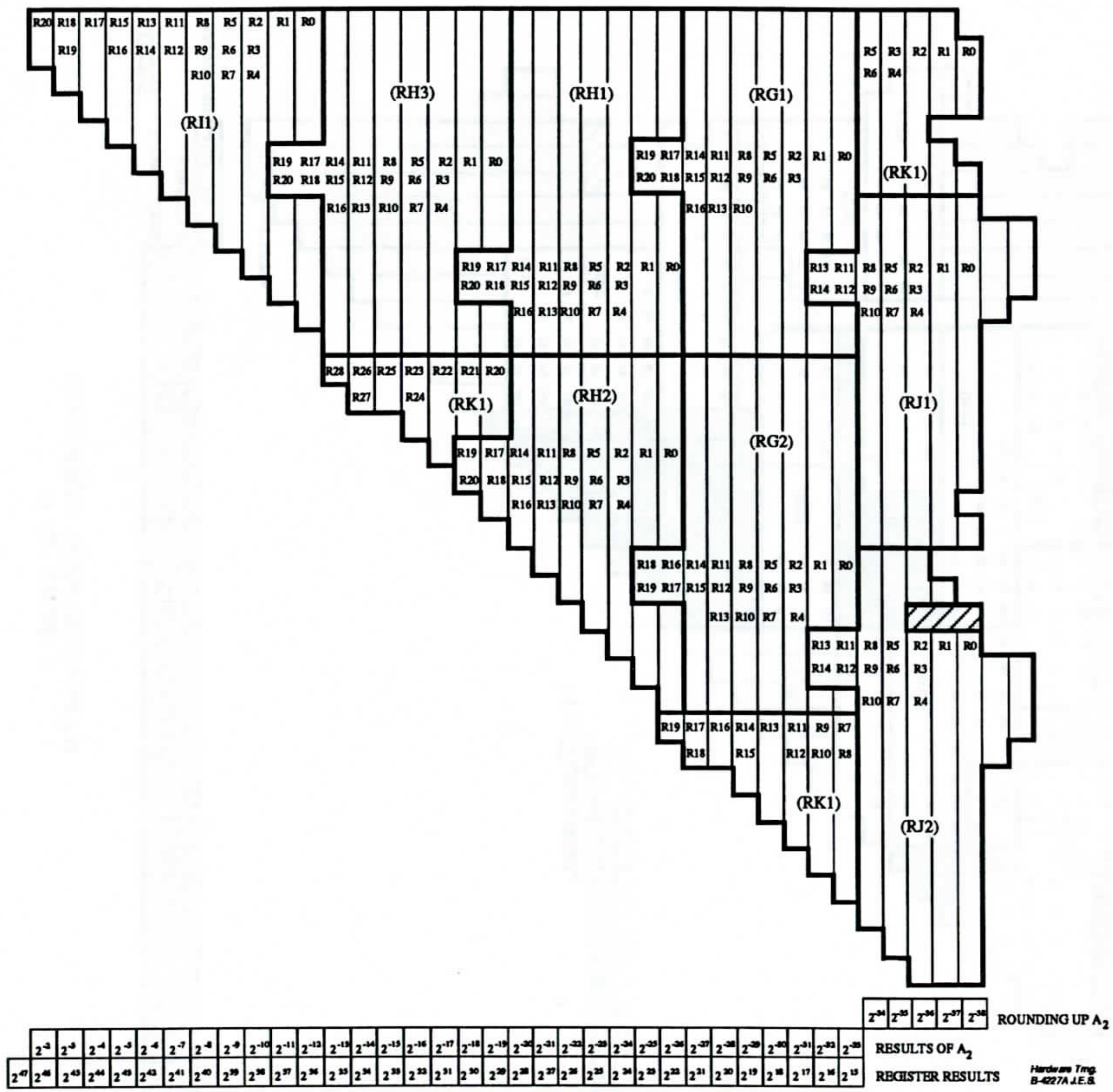
A_1^2 BITS	(RJ1) R TERMS	RECEIVED (RI1)						
2^{-9}	R19	I4						
2^{-10}	R18	I3						
2^{-11}	R17	I2						
2^{-12}	R16	I1						
2^{-13}	R15	I0						

Hardware Trng.
A-4165 J.E.S.

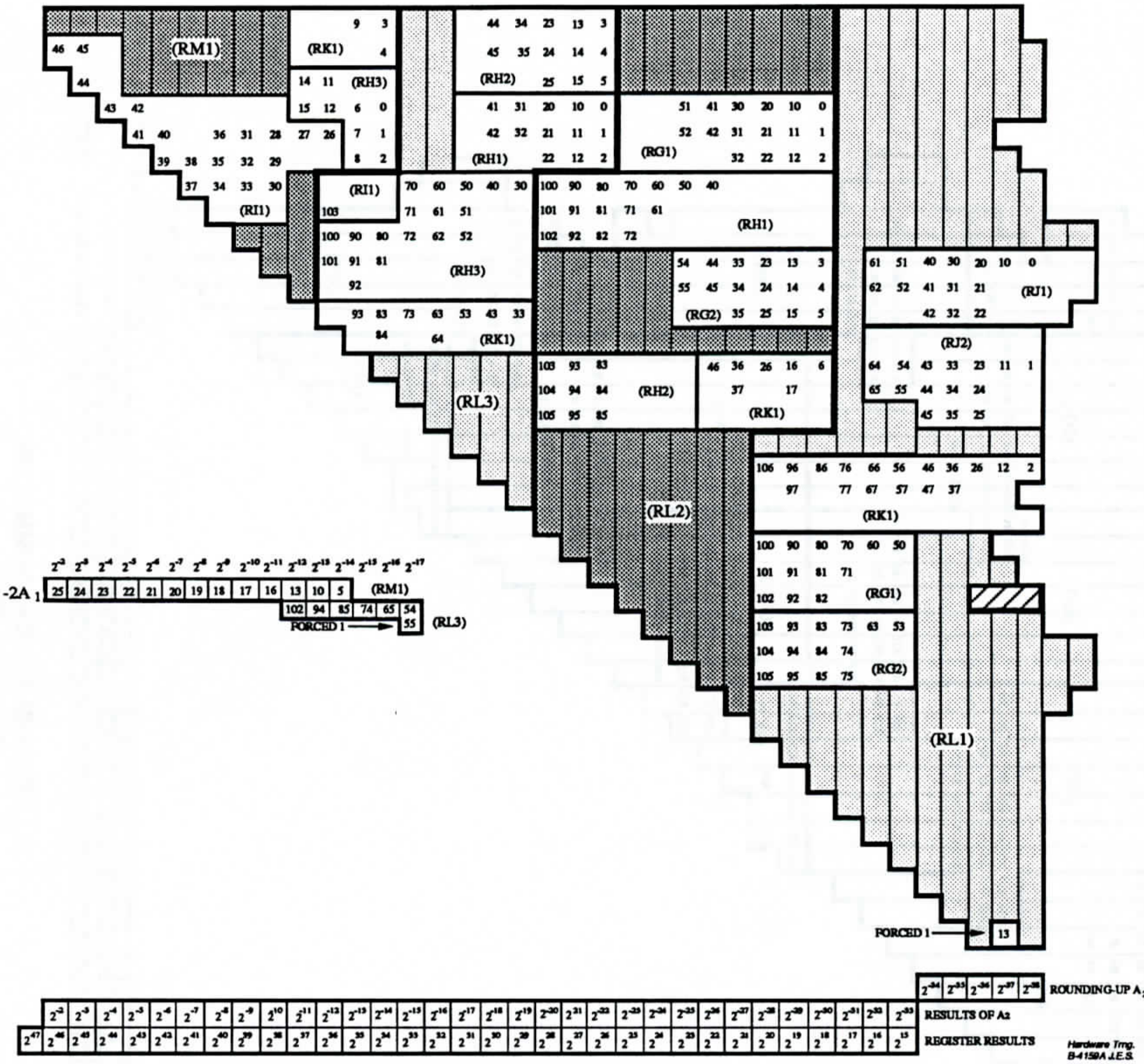
CRAY Y-MP FLOATING RECIPROCAL A_1^2FANOUT
(SHEET 2 OF 2)



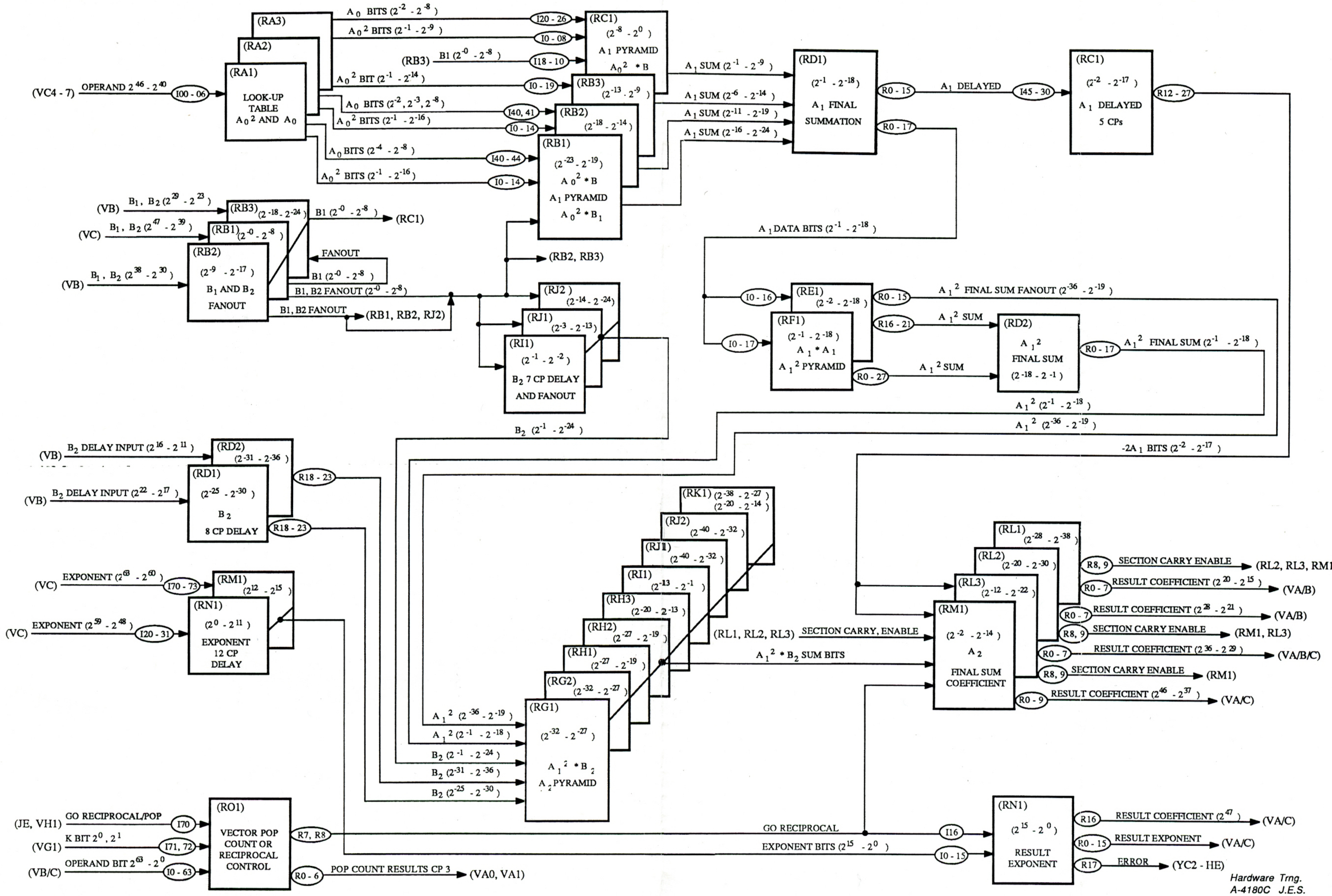
CRAY Y-MP FLOATING RECIPROCAL A_2 PYRAMID
LOGICAL PRODUCTS CP 9



A_2 PYRAMID PARTIAL SUMMATION
R TERMS CP 10



A_2 PYRAMID FINAL SUMMATION
I TERMS CP 11



CRAY Y-MP FLOATING RECIPROCAL BLOCK DIAGRAM