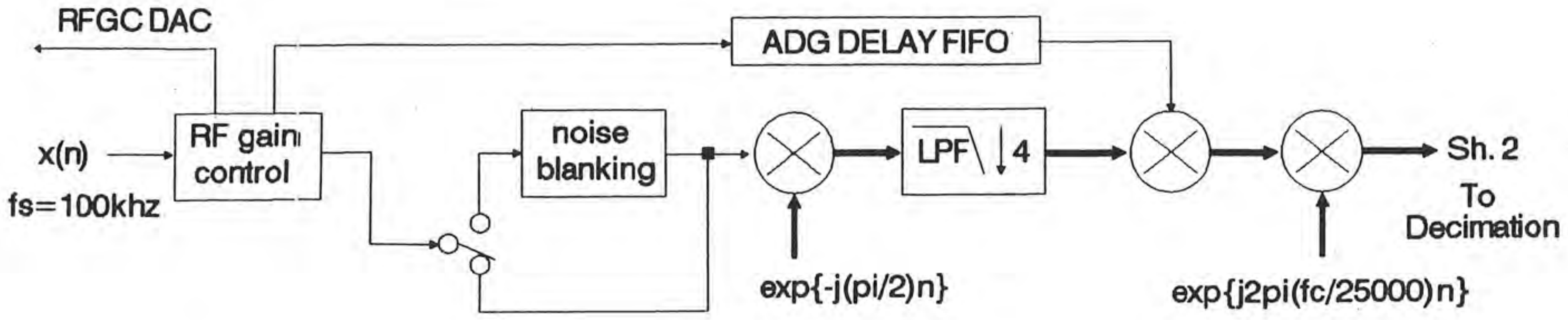
WJ-8711 DSP Block Diagram (Top Level) (Sh. 1 of 4)



fs = sampling rate

fc = fine tune frequency

→ Real Data

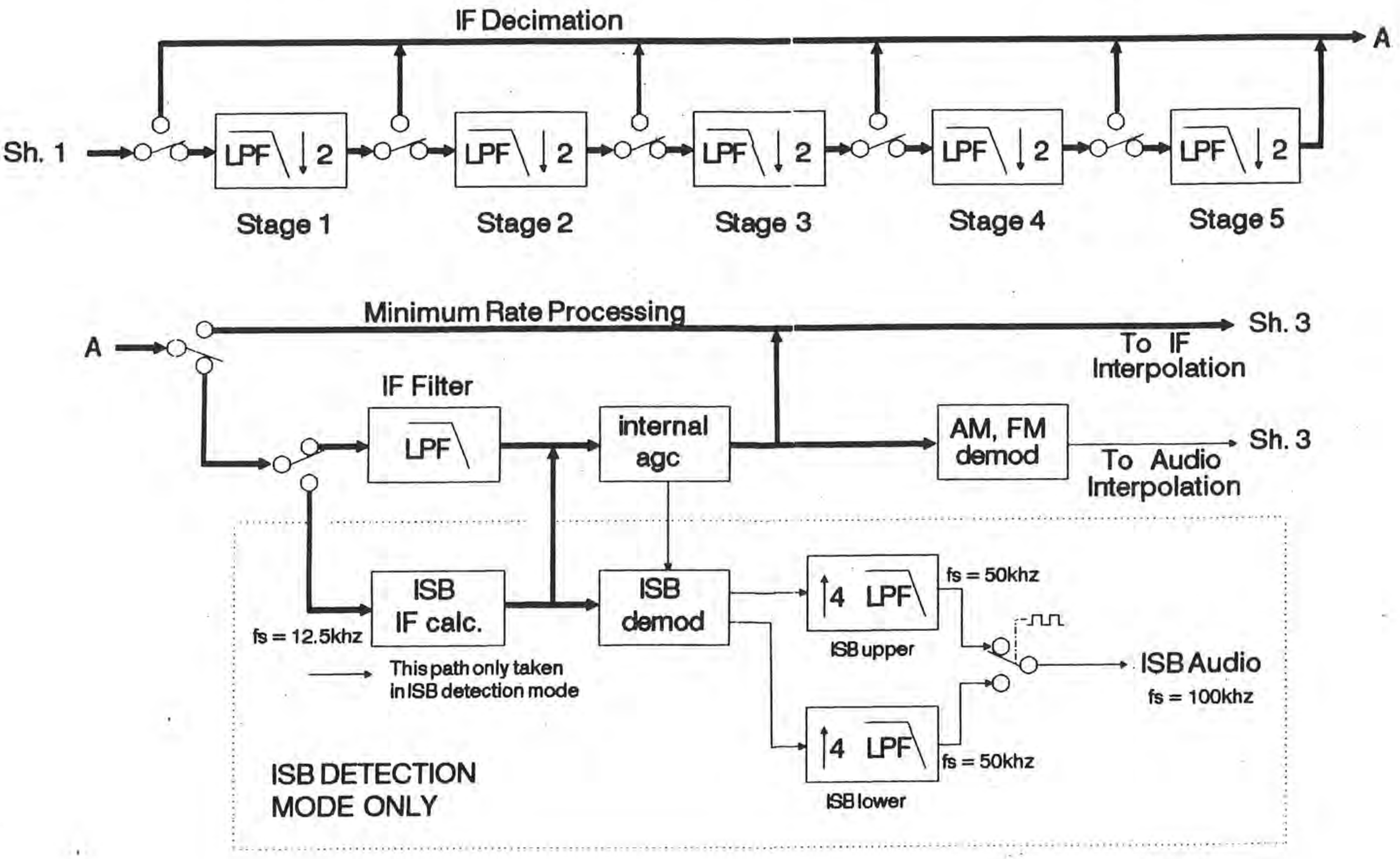
➞ Complex Data

9/30/91

Fig. 1

WJ-8711 DSP Block Diagram (Top Level)

(Sh. 2 of 4)

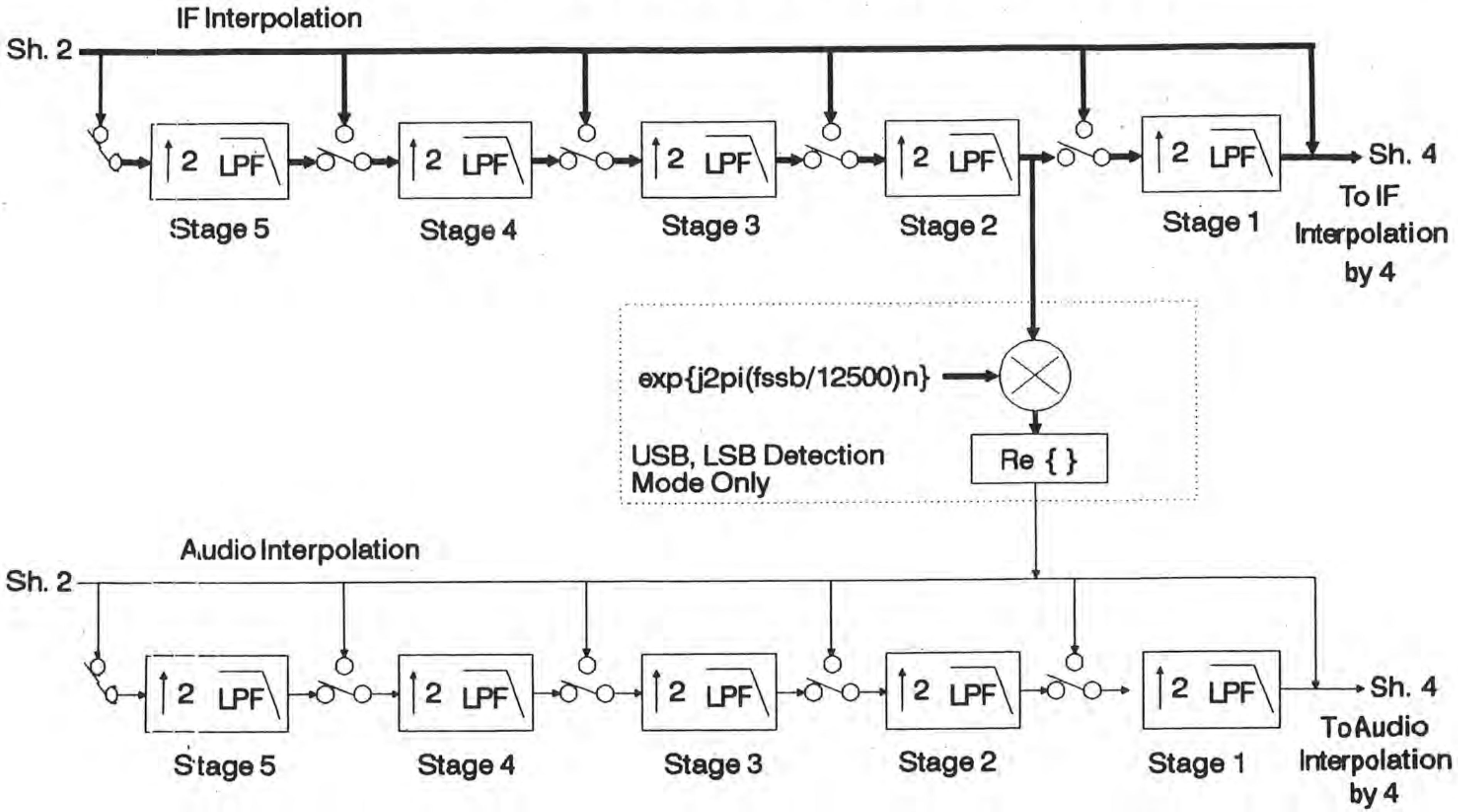


9/30/91

Fig. 1

WJ-8711 DSP Block Diagram (Top Level)

(Sh. 3 of 4)

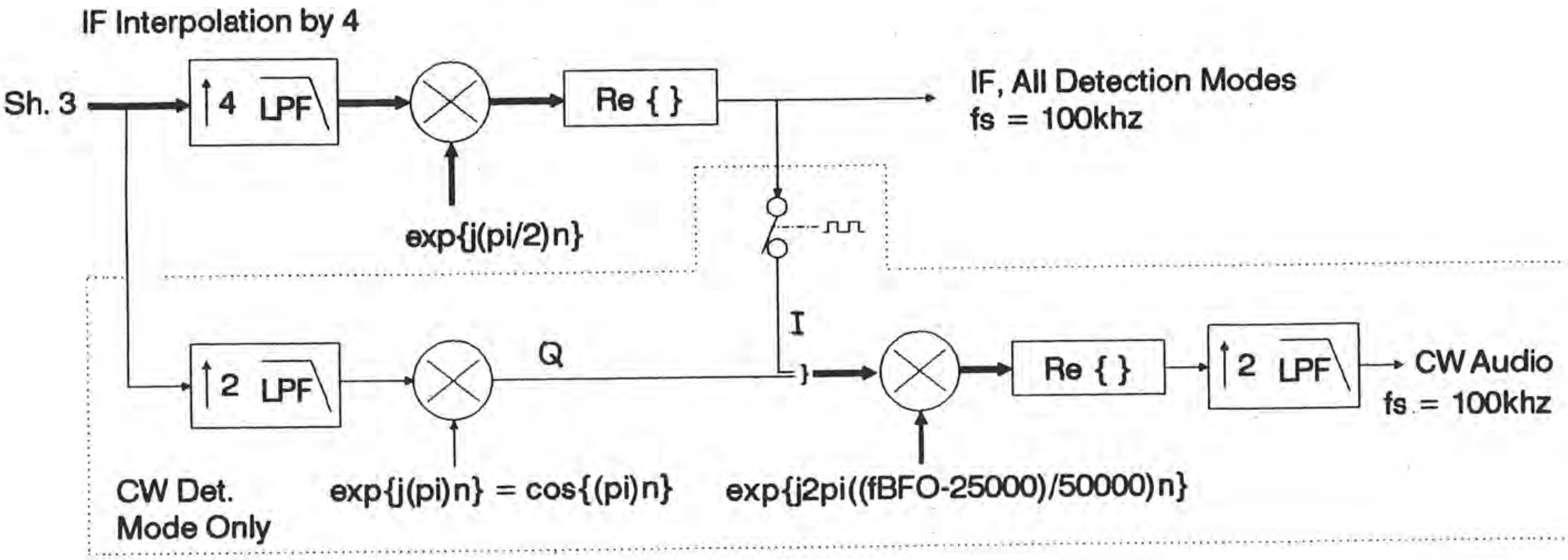


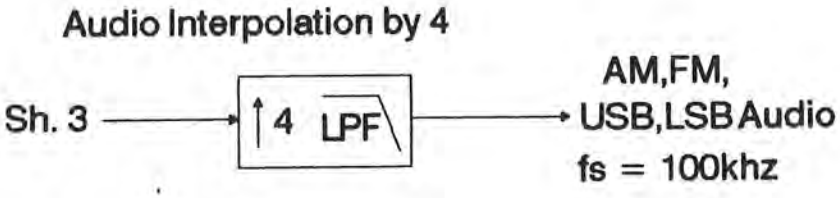
9/30/91

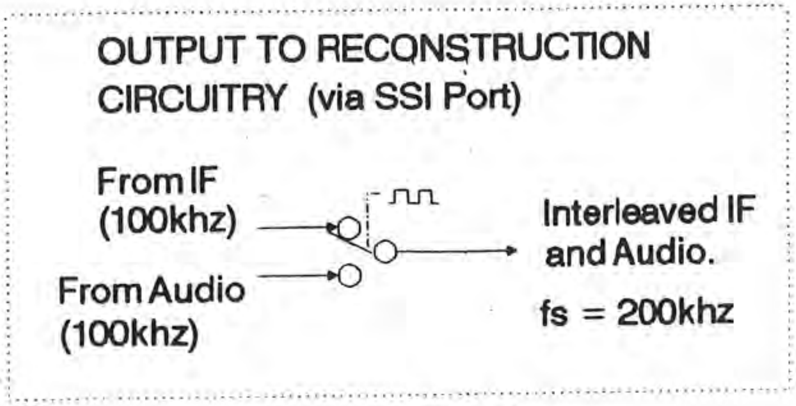
Fig. 1

WJ-8711 DSP Block Diagram (Top Level)

(Sh. 4 of 4)







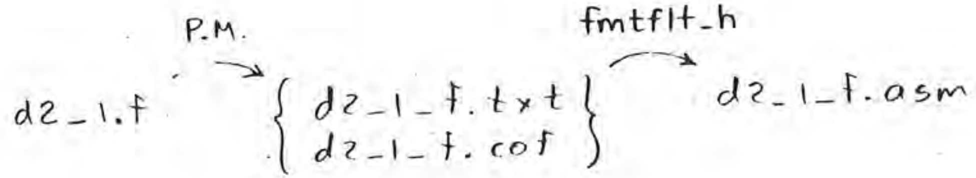
9/30/91

Fig. 1

Digital Filter Summary

Sh. 1 of 4

Decimation/Interpolation by 2



Decimation by 4

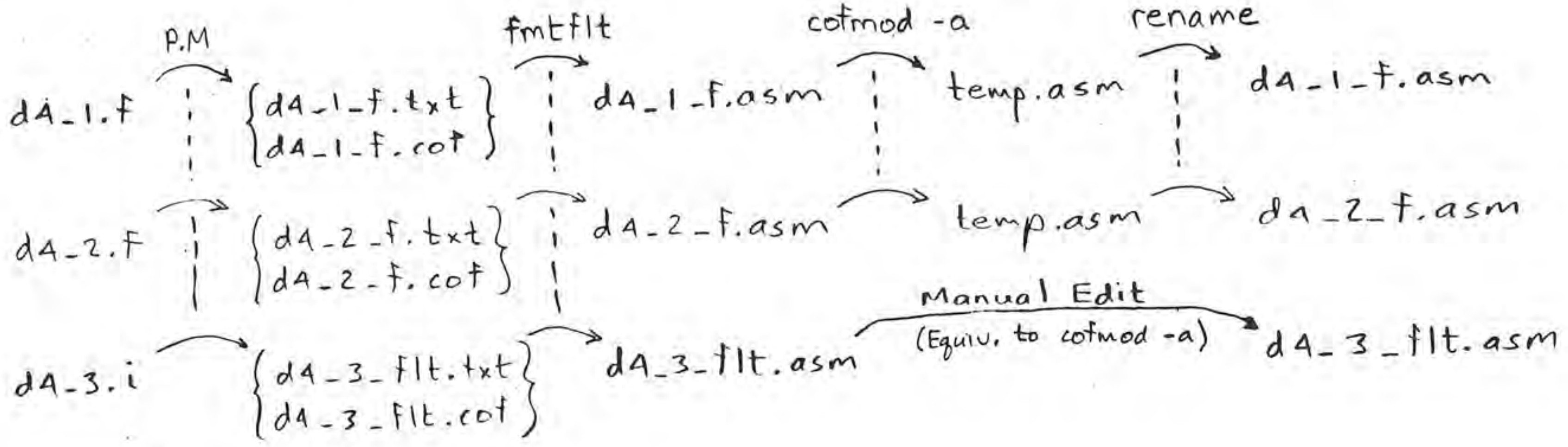


Fig. 16
1 of 4

Digital Filter Summary Sh. 2 of 4

Interpolation by 4 (IF)

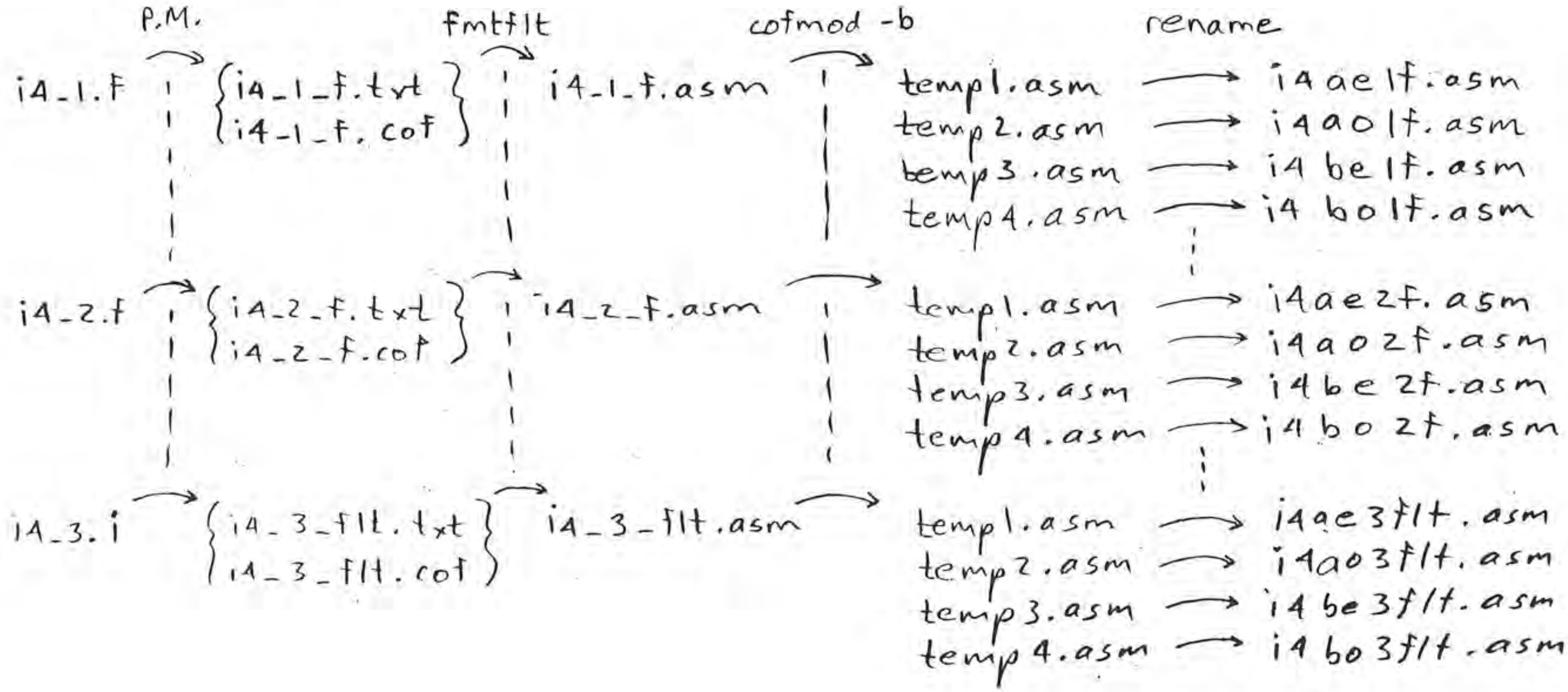
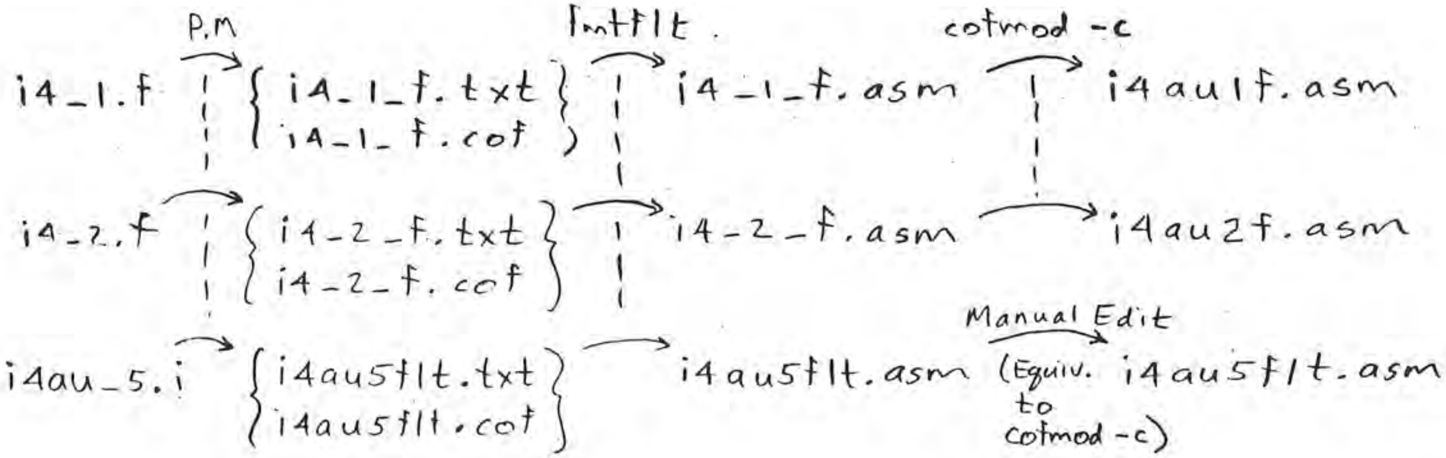


Fig. 16
2 of 4

Digital Filter Summary Sh 3 of 4

Interpolation by 4 (Audio)



IF filters

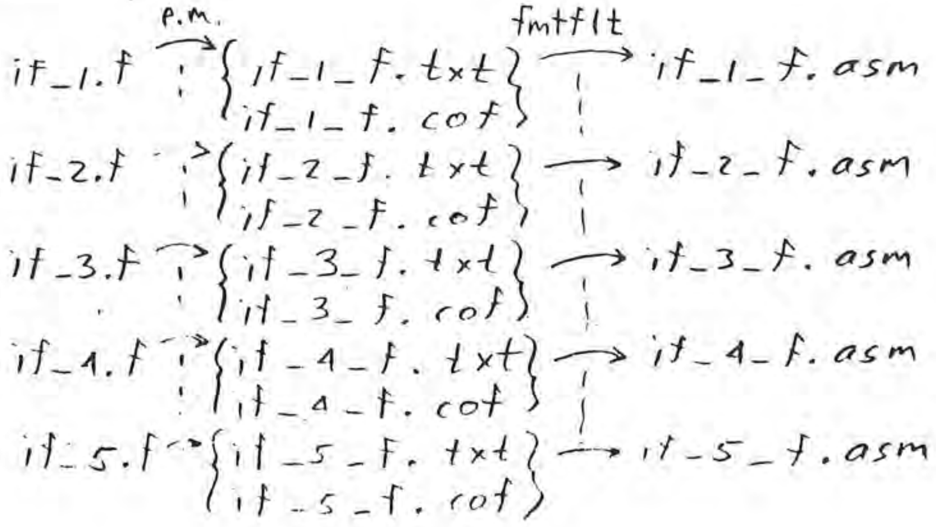


Fig. 16
3 of 4

Digital Filter Summary Sh. 4 of 4

CW Audio Interpolation by 2

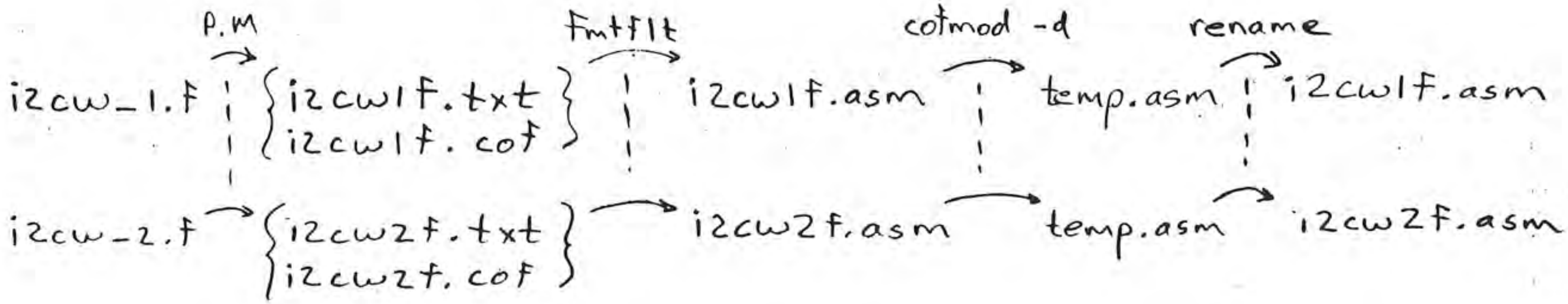
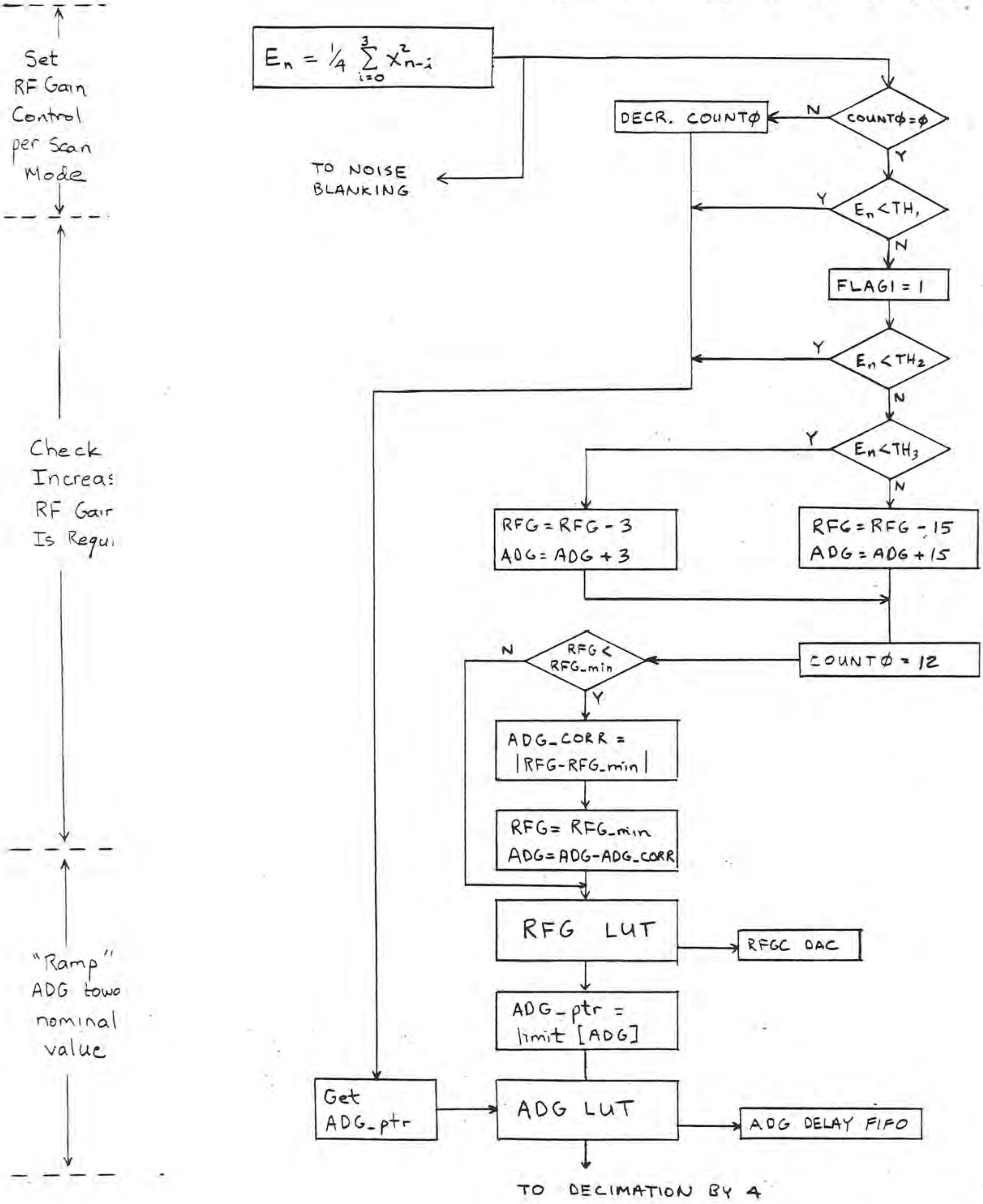


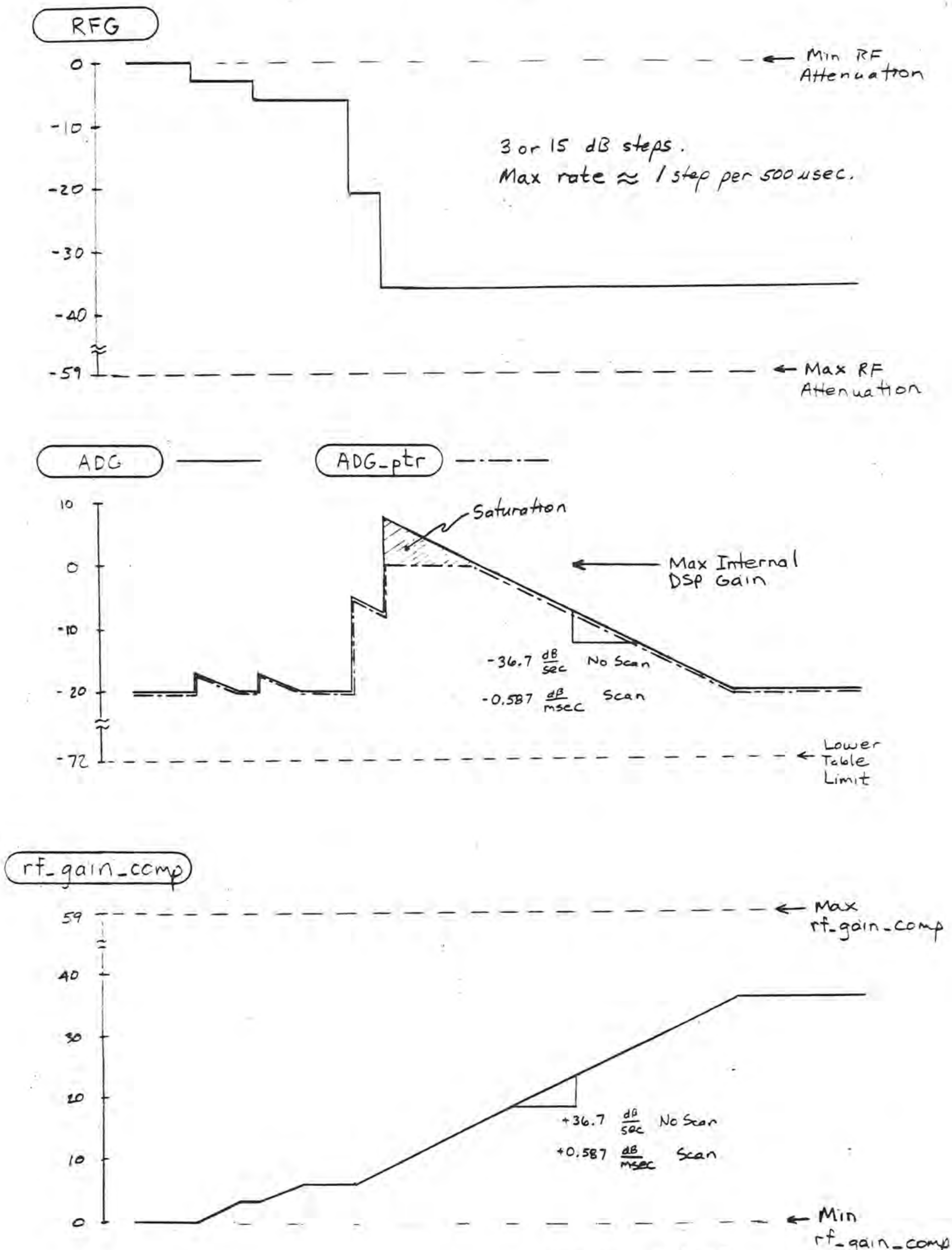
Fig. 16
4 of 4

WJ-8711 RF GAIN CONTROL



9/
F

WJ-8711 Decreasing RF Gain



3 or 15 dB steps.
Max rate ≈ 1 step per 500 usec.

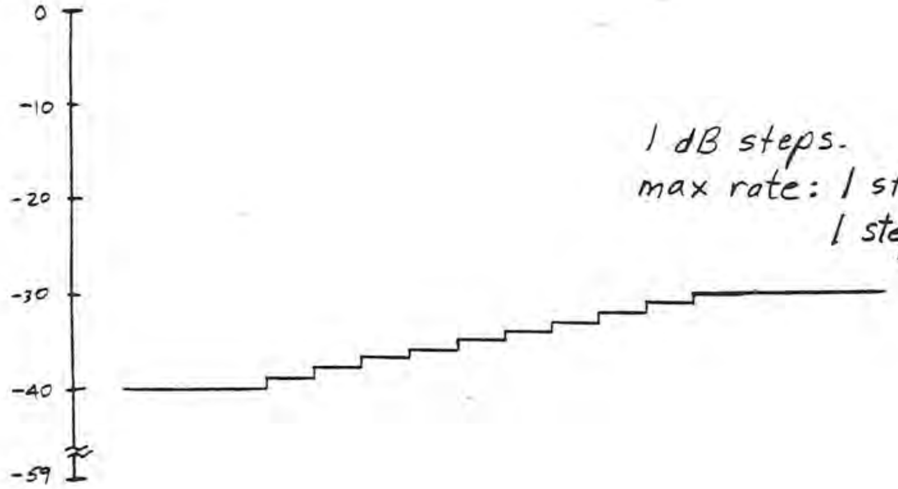
$-36.7 \frac{dB}{sec}$ No Scan
$-0.587 \frac{dB}{msec}$ Scan

$+36.7 \frac{dB}{sec}$ No Scan
$+0.587 \frac{dB}{msec}$ Scan

9/24/91 Fig. 5

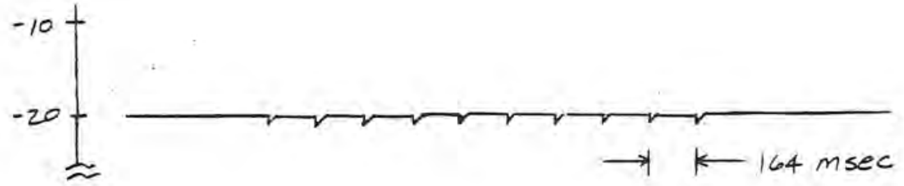
WJ-8711 Increasing RF Gain

RFG



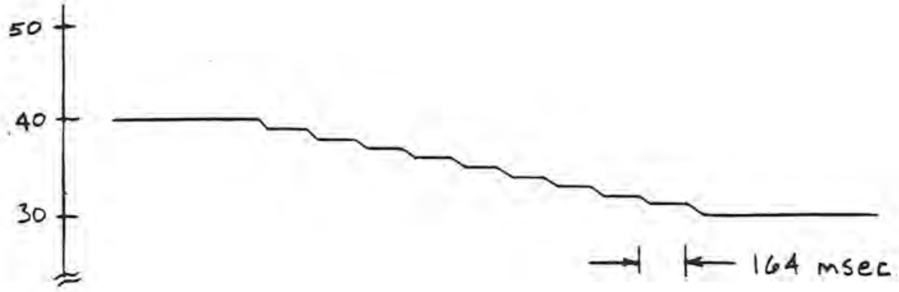
1 dB steps.
max rate: 1 step per 164 msec. Scan off.
1 step per 2.56 msec. Scan on.

ADG = ADG_ptr since saturation doesnt occur in this example.



Scan off

rf_gain_comp



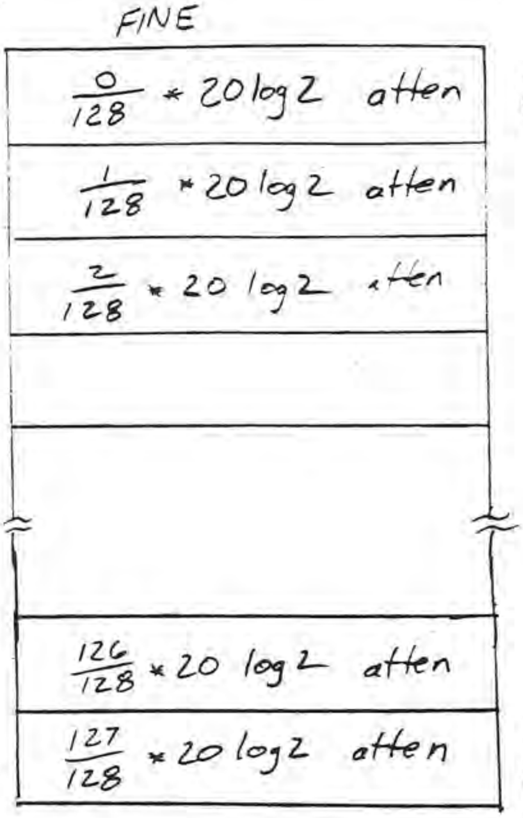
Scan off

9/24/91 Fig. 6

SAVE

Gain Control LUTs

FINE

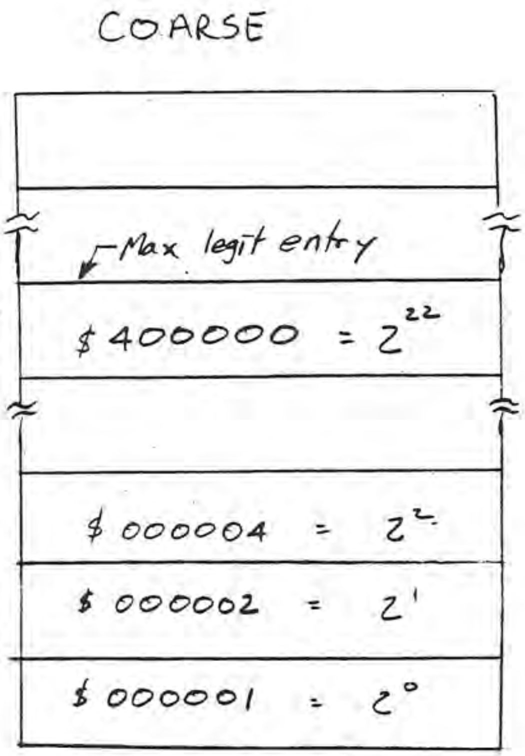


Relative Address		
$7F = 127	$\frac{0}{128} * 20\log 2$ atten	(no atten)
$7E = 126	$\frac{1}{128} * 20\log 2$ atten	
	$\frac{2}{128} * 20\log 2$ atten	less atten. ↑
	$\frac{126}{128} * 20\log 2$ atten	
$0 = 0	$\frac{127}{128} * 20\log 2$ atten	(1 fine step less than 6 dB atten)

Each step is $\frac{20\log 2}{128}$ dB attenuation

Entries shown here are in dBs. The actual table has the linear value ($< x \leq .\overline{9}$)

COARSE



$1F = 31		max gain
	← Max legit entry	
$16 = 22 →	$400000 = 2^{22}	↑ more gain
	$000004 = 2^2	$2 \cdot 20\log 2$ dB
$1 = 1	$000002 = 2^1	$1 \cdot 20\log 2$ dB
$0 = 0	$000001 = 2^0	$0 \cdot 20\log 2$ dB

Transforms smallest possible value of $000001 to $400000. Any more gain would result in overflow.

Fig. 11
1 of 2

Address Generation for LUT

23 22 21 20 19 18 17 16 15 14 13 12 11 8 4 0

$$0\ \underbrace{c\ c\ c\ c\ c}_{5\text{ bit}}\ \underbrace{f\ f\ f\ f\ f\ f\ f}_{7\text{ bit}}\ x\ x\ x\ x\ x\ x\ x\ x\ x\ x\ x$$

Net gain = 5 bit coarse address * 20 log 2

$$\text{minus } \frac{\left[127 - 7\text{bit fine address}\right]}{128} * 20 \log 2 \quad \text{dB}$$

Example: Address = $0\ \underbrace{1\ 0\ 1\ 1\ 0}_{5\text{bit}}\ \underbrace{0\ 0\ 0\ 1\ 0\ 1\ 1}_{7\text{ bit}}\ x\ x \ldots$

$$\text{Gain} = 22 * 20 \log 2 - \frac{(127-11)}{128} * 20 \log 2$$

Gain = 126.997 dB

Fig. 11
2 of 2

Courtesy of http://BlackRadios.terryo.org

Manual Gain

Save =
Manual Gain
approach.

This equation relates the gain produced from the two lookup tables to the value required to "feed" the LUT address generation code.

$$\text{Most significant bits} = \text{lsr}\left[\left(\text{Gain}/\tfrac{20\log 2}{128}\right) + 127\right] \leftarrow \text{Exact.}$$

$$\text{Top 3 Hex digits} \approx \left[\left(\text{Gain}/.047\right) + 127\right]/2$$

Min Gain = -41.4 → ...FF D0F

1110110000

1110100001111 = $E87800

Max Gain = 96.0

Top 3 Hex digits ≃ 43D → 430000

Table **Min Gain →** $E87800 + 0 * 23 * ($800)

$E87800 + 1 * 23 * ($800)

$$\$E87800 + 2 * \underbrace{23 * (\$800)}_{} \rightarrow \text{Step size}$$

⋮

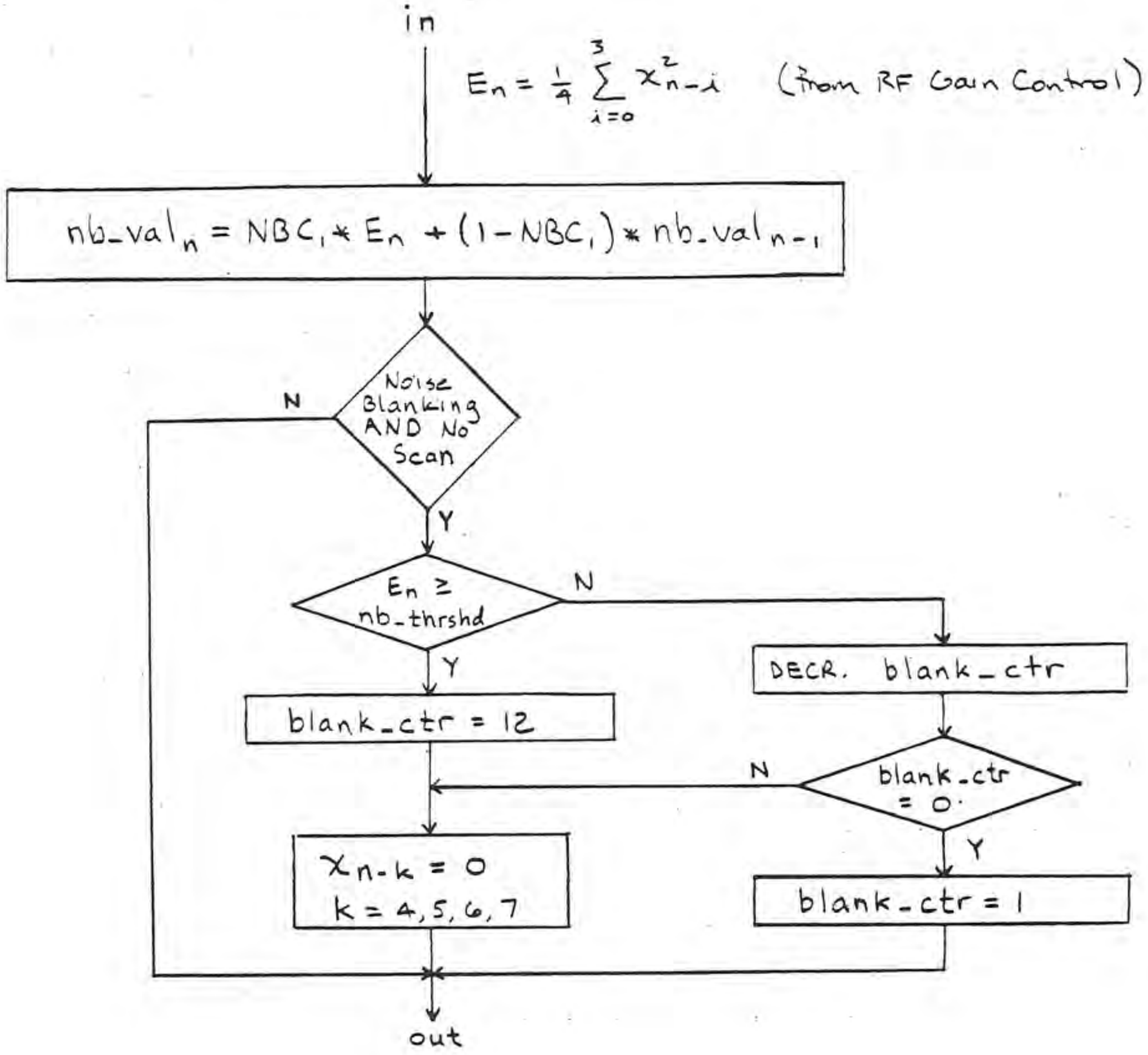
$E87800 + 127 * 23 * ($800)

NORM 3EA

ATTEN 4D2

Fig. 12

WJ-8711 Noise Blanking



P/O IRQB INTERRUPT SERVICE ROUTINE

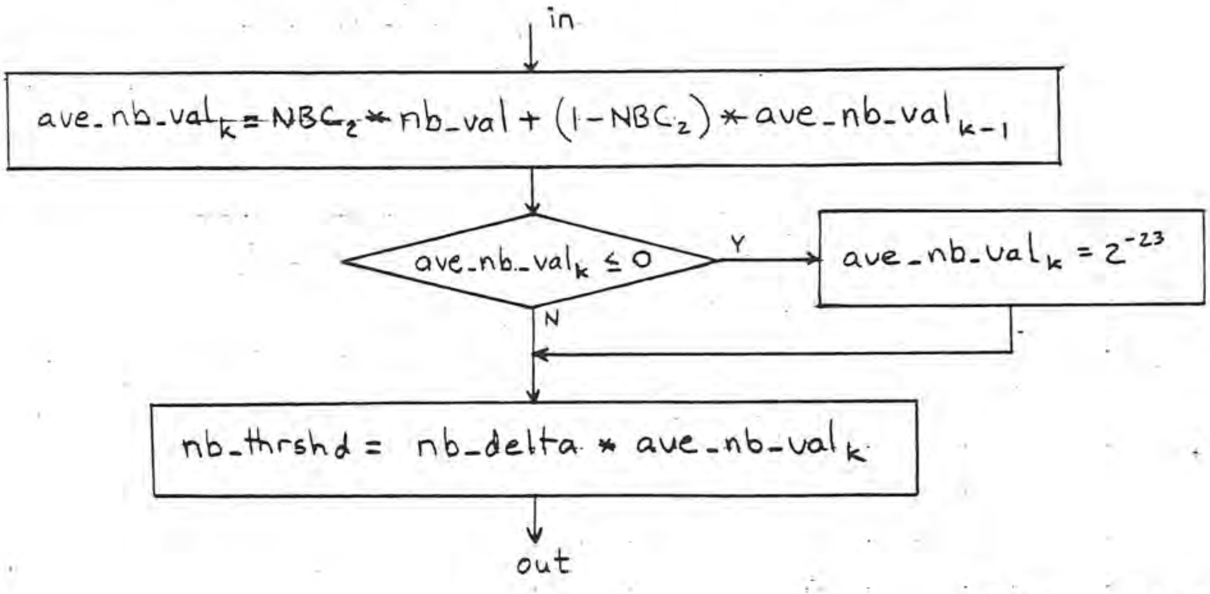
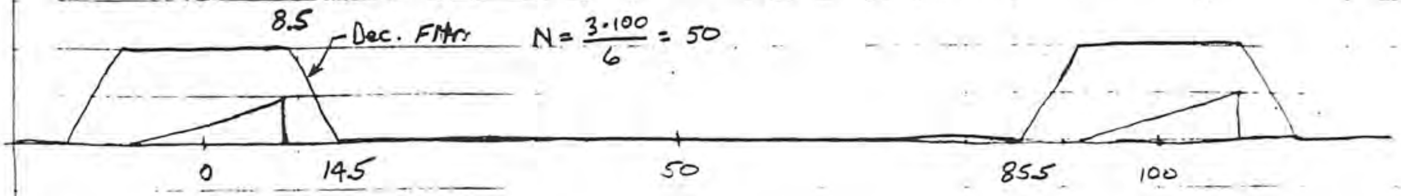


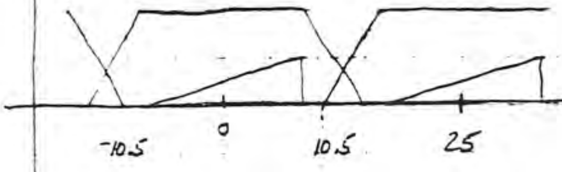
Fig. 7 9/24/91

← THIS IS WHAT WILL PROBABLY BE USED.

Computations – Approach 2 (Dec. by 4)

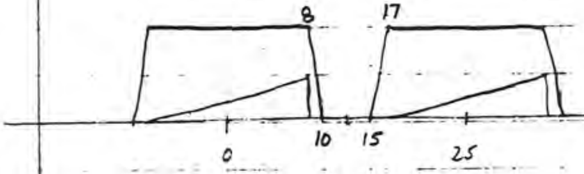


$N = \frac{3 \cdot 100}{6} = 50$

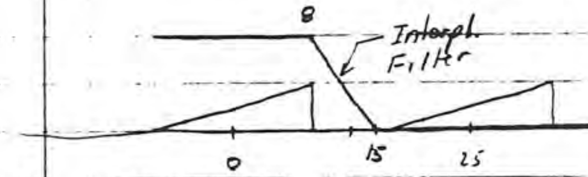


After decimation by 4

Fine Tune before Channel Filter



Channel Filter $N = \frac{3 \cdot 25}{2} = 37.5$



$N = \frac{3 \cdot 100}{7} = 43$

Decimation → $\frac{50}{4} = 12.5$

Ch. Filter → $\frac{75}{4} = 18.75$

Interpolation → $\frac{43}{4} = 10.75$

42

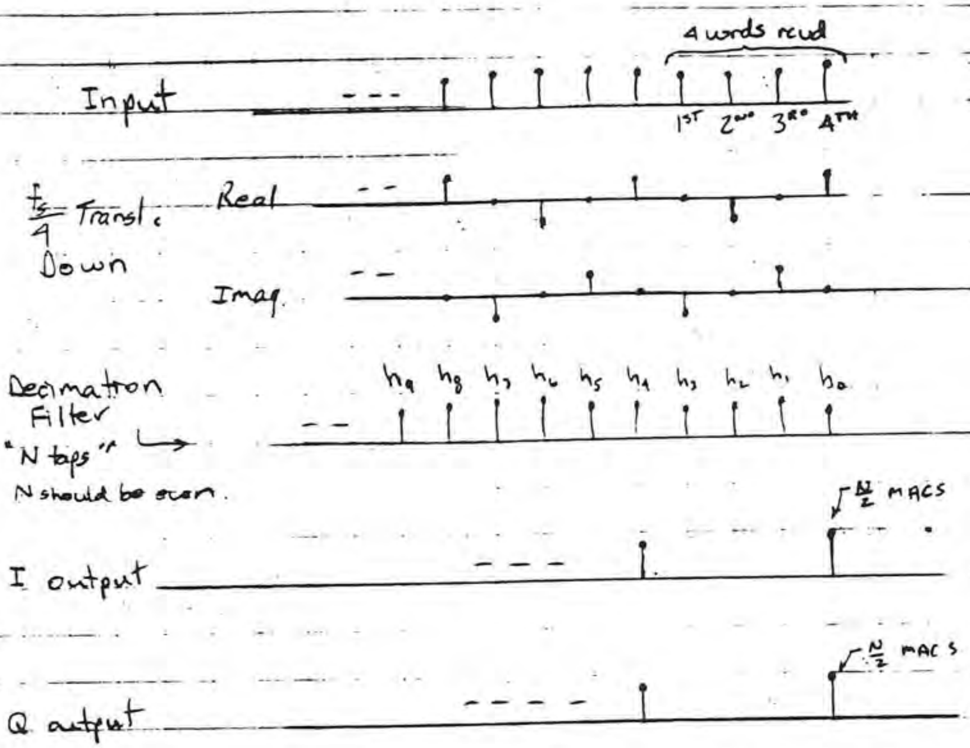
Intpl Filter could be the same as dec. filtr to save storage. ~~cost would be~~ $\frac{7}{4}$ Instr.

However, filters are ~~stored differenty~~ to minimize overhead

Note: These filters were tweaked a little so the ~~frequencies~~ shown are not exact. See filter files for exact values.

Fig. 8

Decimation by 4 with $f_s/4$ Translation Down

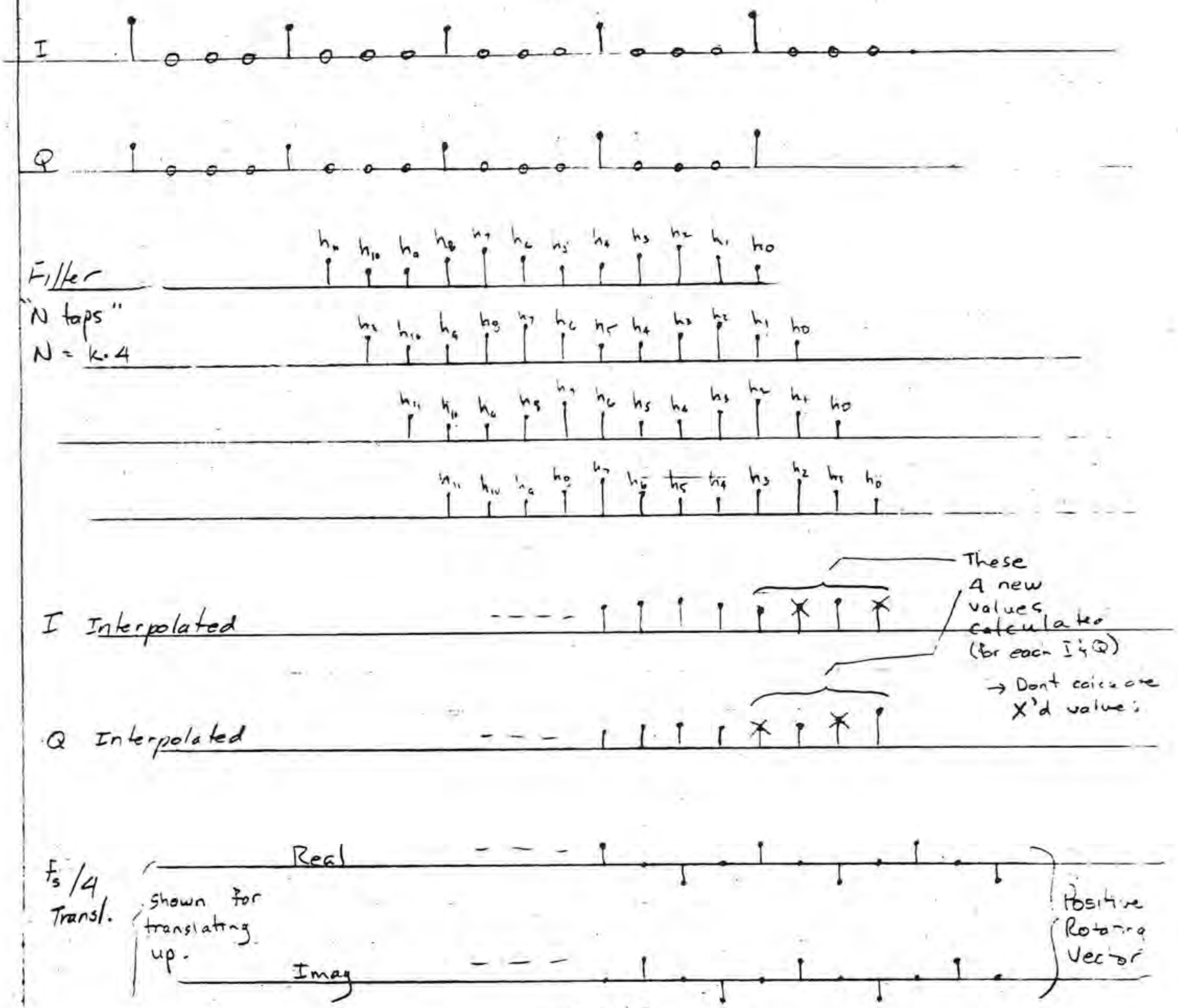


Sign change on coefficients to translate down:

h_0	normal
h_1	normal
h_2	invert
h_3	invert
h_4	normal
h_5	normal
h_6	invert
h_7	invert
⋮	

Fig. 9

Interpolation by 4 w/ $\frac{f_s}{4}$ Translation Up



$$\text{Output} = \text{Real}\left\{\overbrace{(I + Qj)}^{\text{Interpolated Data}}\overset{\swarrow f_s/4\ \text{transl.}}{(\text{Real} + \text{Imag}\, j)}\right\}$$

NOTE CHANGE

$$= I \cdot \text{Real} - Q \cdot \text{Imag}$$

↑— Invert the Imag to accomplish this negate

Sign change to accomplish $\frac{f_s}{4}$ up translation & taking real part

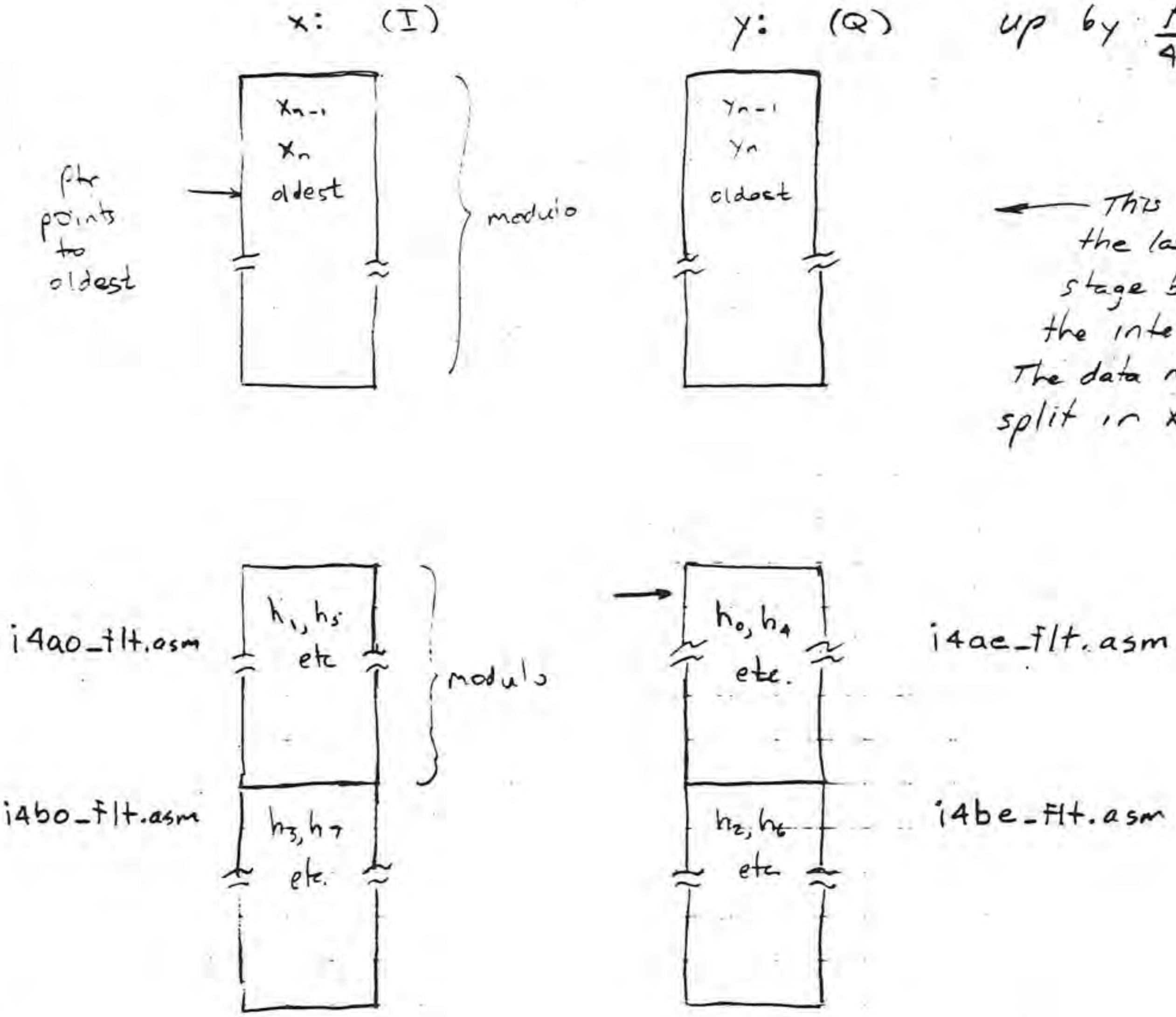
		CHANGED TO
h_0	normal	normal
h_1	invert	normal
h_2	invert	invert
h_3	normal	invert
h_4	normal	normal
h_5	invert	normal
h_6	invert	invert
h_7	normal	invert

for upright IF spectrum. (required $f_s/4$ translation DOWN versus original UP)

↳ UP would have worked except analog IF was inverted spectrum

Fig 10
1 of 2

Buffer & Filter Structure for Interpolation with Translation up by $\frac{F_s}{4}$.



This is the last stage before the interpl. The data must be split in x: & y:

NOTE: This still requires an address reg. update after calculating two values.

If I & Q were interleaved in x: and coefficients were in y:, the extra overhead would be:

	Instr
load n reg. with #2.	1
update coeff ptr, data ptr.	3

Fig 10 (cont.) 2 of 2

Sig Strength Reporting

Intgn_gn = -29 - X dBm

This is the required integration gain to bring an X dBm signal up to the set point (Normal RF input)

Intgn_gn = -29 - (-135) = 106
⋮
Intgn_gn = -29 - (+20) = -49

Signal Level	Intgn Gain (-15 dB Atten)	Intgn Gain (Normal)	Intgn Gain (+10 dB)
-135	106 + 15	106	106 - 10
-134	105 + 15	105	105 - 10
-133	104 + 15	104	104 - 10
⋮	↑	⋮	↑
-30		1	
-29	NOTE: Since the integrator is 15 dB higher, 15 has to be subtracted to get the real representation of the signal level	0	NOTE: same thing here. Add 10 dB to the intgr output.
-28		-1	
-1		-28	
0		-29	
1		-30	
2		-31	
⋮			
20		-49	

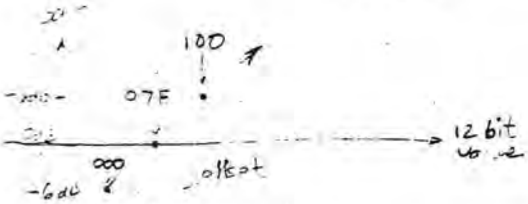
This value exists in the code in IRQB

Fig. 15
1 of 3

Sig Strength Reporting

The output of the integrator has the following format.

$$\underbrace{|\,s\;c\;c\;c\,|\,c}_{5\text{ bit}}\;\underbrace{c\;f\;f\,|\,f\;f\;f\;f\,|\,f}_{7\text{ bit}}\;x\;x\;x\,|\,x\;x\;x\;x\,|\,x\;x\;x\;x\,|$$



See next page

This bit represents $\frac{20 \log 2}{4} = 1.5051\,dB$

Really (1.5051 – 5.97) dB due to offset. See next page.

The integrator (for Normal RF input) must range from +106 dB to -49 dB. This requires 7 bits of magnitude plus the sign bit.

Algorithm: (Start w/ the intgn value that already has RF input accounted for)

① Mpy the intgn gain value by 1.505 so the indicated bit above represents 1.0 dB.

16 bit shift → ② Mpy the result by a shift constant which will shift the result so the magnitude occupies bits [6:0] (A 16-bit shift right)

(1.5051 d)($000080)

.000022966

{Combine ① & ②} Do an mpyr

$0000C1

↑ COMPOSITE MULTIPLIER for ① & ②

③ The ~~max~~ Intgn-gain for normal RF input is ~~106~~ for -135 dBm signals
This will give a 0 value for the Host.

Negate the result of ①&② and add ~~106~~

#> ~~106~~ 5

See next page.

Limit the range from 0 to 155

Fig. 15
2 of 3

Sig Stren Reporting

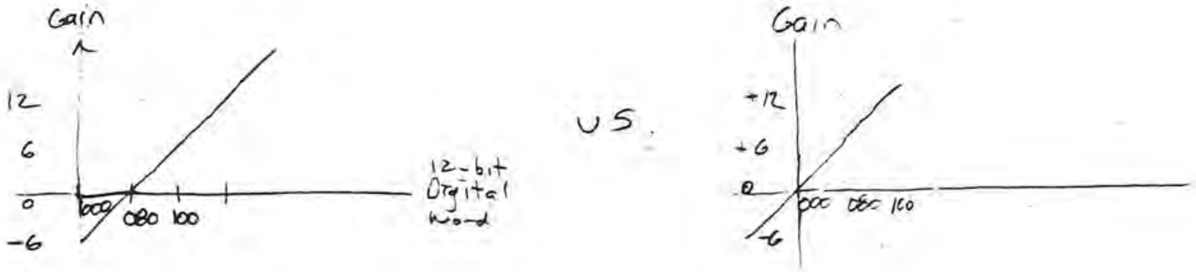
SH 3 OF 3

12 bit code: coarse	fine	dB
0 0 0 0 0	0 0 0 0 0 0 0	-5.97 dB
0 0 0 0 0	0 0 0 0 0 0 1	-5.93 dB
0 0 0 0 0	0 0 0 0 0 1 0	-5.88 dB
0 0 0 0 0	1 1 1 1 1 1 1	0 dB
0 0 0 0 1	0 0 0 0 0 0 0	6.02 - 5.97
0 0 0 0 1	0 0 0 0 0 0 1	6.02 - 5.93
0 0 0 0 1	0 0 0 0 0 1 0	6.02 - 5.88
0 0 0 0 1	1 1 1 1 1 1 1	6.02 dB

128 steps of 20log 2/128

7 bits
representing magnitude of the output of the integrator.

This illustrates the fact that the gain vs 12-bit digital word has an offset of about 6 dB



A

This can be compensated for by adding 112 in the algorithm vs 106.

RESULTS: I checked this out and it appeared to be working (to within about a dB) over the range -130 to +10
I didn't go beyond those limits.

Fig. 15
3 of 3

The desirable format for running the programs is

prog1 <arg1> <arg2>

with <arg1>.asm being a user specified source file and <arg2>.asm being a user specified dest file.

The same is true for the 3RD & 4TH cases.

The 2ND case where there are 4 output files could just append 1, 2, 3, 4 to <arg2> so the 4 output files would be

<arg2>1.asm	h0, h4..
<arg2>2.asm	h1, h5..
<arg2>3.asm	h2, h6..
<arg2>4.asm	h3, h7...

The four cases on the previous sheet are options -a, -b, -c, -d respectively;

The program is cofmod.c.

Fig. 17
2 of 2

WJ8711 – DSP EPROM

EPROM organization.

→ The address in the .LOD file is 1/3 the EPROM address where the data will be stored.

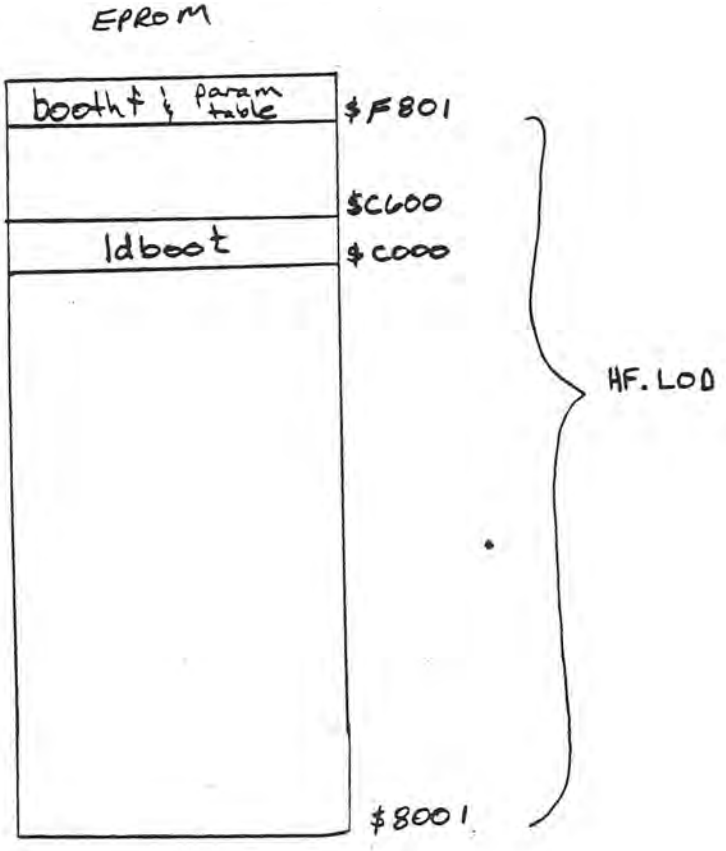


Fig. 18
2 of 2