128/64-point FFT/IFFT Module
Design Description
Magis Doct # E10029

Version 1.9

19 January 2002

Table of Contents

1 Introduction

The DFT of a finite-length sequence of length N is

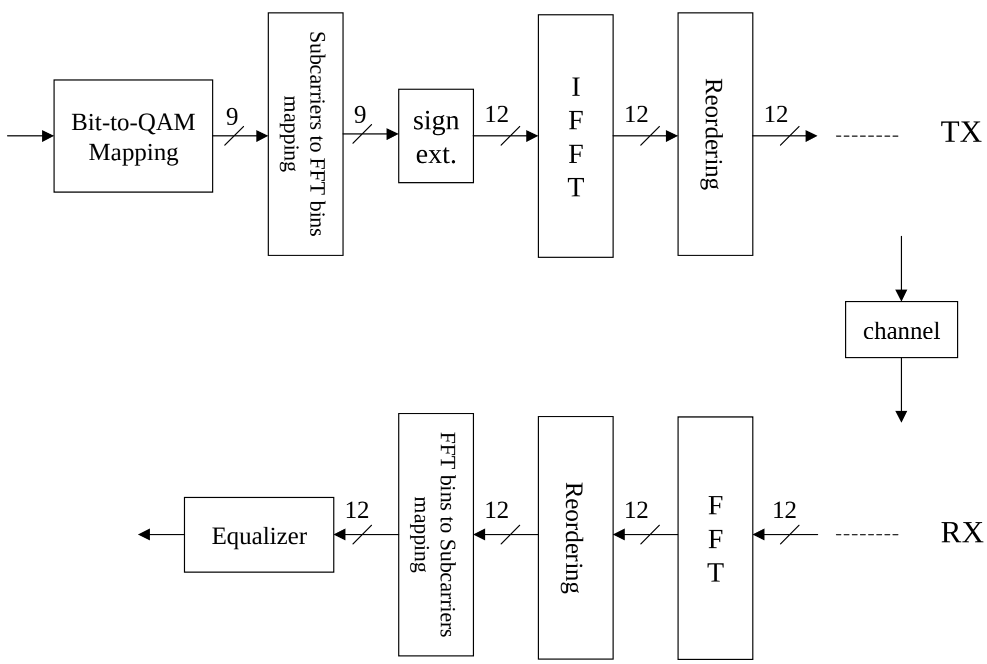
$$X[k] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad (1)$$

where $W_N = e^{-j(2\pi/N)}$. The inverse DFT is given by

$$x[n] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} X[k] W_N^{-kn} \quad (2)$$

Since equations (1) and (2) differ only in the sign of the exponent of W_N, the computational procedures for evaluation (1) apply to (2) with straightforward modification of only the twiddle factor exponent signs. This allows us to use the same ASIC block to do both the FFT (in the receiver) and the IFFT (in the transmitter). Figure 1 shows the configuration of the FFT and IFFT in the OFDM system. In fixed-point computation however, the dynamic ranges involved for the FFT and IFFT may differ and care must be taken to include this factor.

Figure 1: FFT/IFFT in OFDM Transceiver



References:

[1] E. Word and A. Despain, "Pipeline and parallel-pipeline FFT processor for VLSI implementation". IEEE Trans. Computer, May 1984.

1.1 Revision History

Revision	Date	Author	Comment
0.0	20 September 2000	B. Hu	Draft Version
1.0	3 October 2000	B. Hu	Major modifications: 1) Swapped the sections of 128-FFT and 64-FFT to emphasize 128-FFT; 2) Changed the bit widths and added internal saturation. 3) Added Figure 3 to detail the bit width in each stage. 4) Removed the 20MHz clock option; 5) Listed the control signals in Section 2.3; 6) Added MUX control signal for stage 7 in Figure 3; 7) Added Section 5.2 of quantization effect on out-of-band radiation; 8) Added a new section of test vector and provided 3 test vectors.
1.1	13 October 2000	B. Hu	Corrected two typing errors: 1) In Figure 3 and 4, should be "round off 9 LSB" after the multiplier; 2) The titles of Table 3 and 4 should be of 64-FFT.
1.2	17 October 2000	B. Hu	1) Saturated twiddle factors at ±511; 2) Replaced all rounding with truncation; 3) Changed the test vectors and simulation results correspondingly. 4) Added test vectors for 64-pt FFT
1.3	24 October 2000	B. Hu	Added subcarriers-to-FFT bins mapping.
1.4	30 April 2001	H. Mai	Update Pisces implementation details.
1.5	2 May 2001	H. Mai	Correct signal names to match fft module names. Reference readme file for testbench info.
1.6	29 May 2001	J. Wang	Audit. Added a section on future

			improvement.
1.7	13 October 2001	B. Hu	Corrected the figures numbering.
1.8	11 January 2002	L. Huynh	Added Audit Section
1.9	19 January 2002	J.A. Crawford	Close out audit.

1.2 Acronyms

Acronym	Translation
IFFT	Inverse Fast Fourier Transform
FFT	Fast Fourier Transform

1.3 Open Issues List

Revision	Item	Status/[Assignee]

2 Interface Signals

2.1 Data

Data Signal	I/O	# bits	Speed/Timing	Comment
Cmap_i[11:0]	I	12	40 MHz	IFFT/FFT in-phase input.
Cmap_q[11:0]	I	12	40 MHz	IFFT/FFT quadrature-phase input.
Cmap_data_valid	I	1	Burst	IFFT/FFT data valid input.
fft_i[11:0]	O	12	40 MHz	IFFT/FFT in-phase output.
fft_q[11:0]	O	12	40 MHz	IFFT/FFT quadrature-phase output.
Fft_data_valid	O	1	Burst	IFFT/FFT data valid output.

2.2 Clocks

Data Signal	I/O	# bits	Speed/Timing	Comment
Clk	I	1	40 MHz	40 MHz clock
Clk_en	I	1	20 MHz	20 MHz clock enable.

2.3 Control

Data Signal	I/O	# bits	Speed/Timing	Comment
Reset_n	I	1	-	Reset input.
Init	I	1	Burst	Synchronous reset input 1 = initialize
Guard_p8_sel	I	1	Burst	Guard interval select. 1 = 0.8us guard interval 0 = 0.4us guard interval
Fft64_sel	I	1	-	64 point IFFT/FFT select 0 = 128 point IFFT/FFT 1 = 64 point IFFT/FFT
Ifft_sel	I	1	-	Inverse FFT select 0 = FFT select 1 = IFFT select

2.4 Test

Data Signal	I/O	# bits	Speed/Timing	Comment

3 ARM Register Interfaces

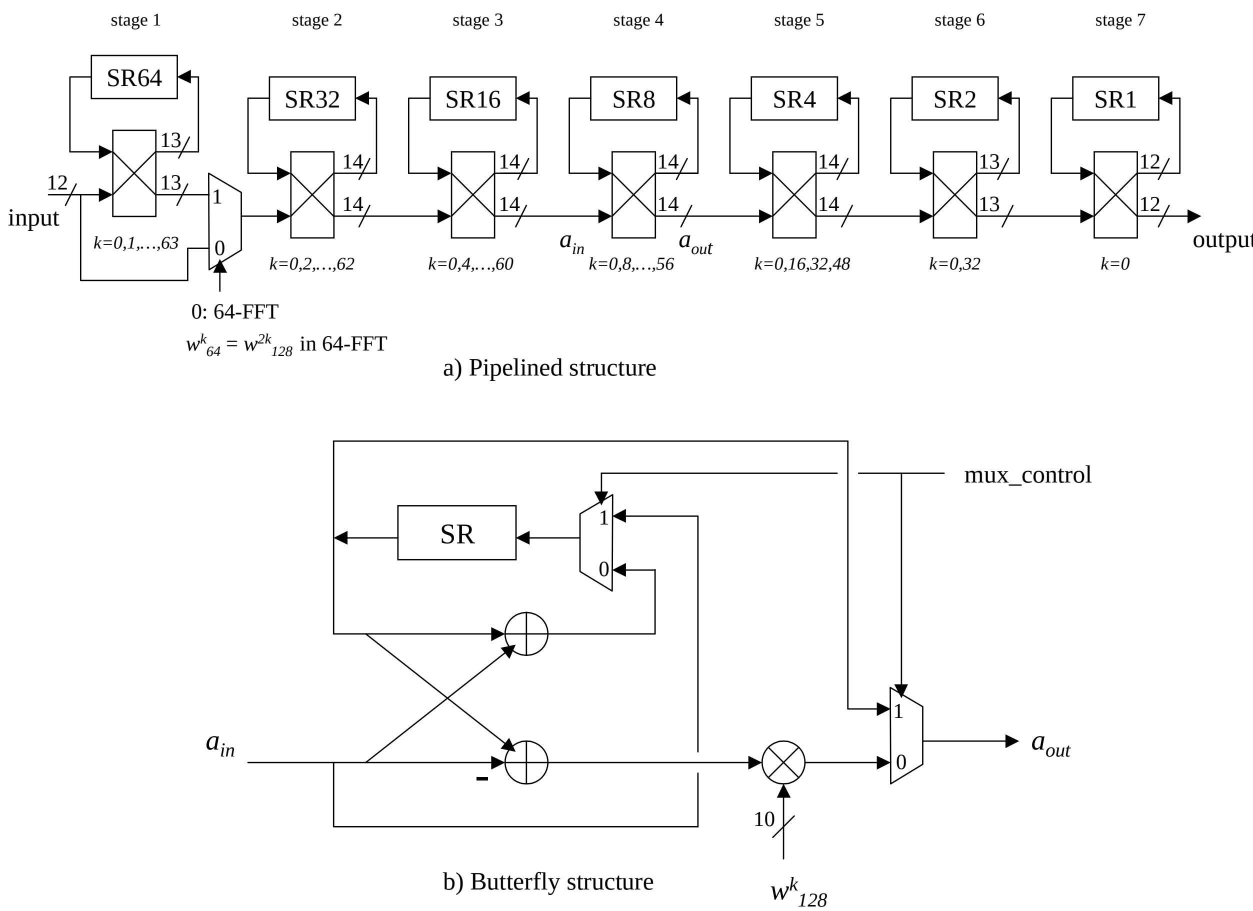
Reference Magis document “E10070 ARM Register Map.doc” for all register descriptions.

4 Design Description

This design is based on the Radix-2 Single-path Delay Feedback pipeline structure[1], with decimation-in-frequency algorithm.

4.1 Block Diagram of 128-point FFT

Figure 2: Pipelined 128-FFT Implementation Details



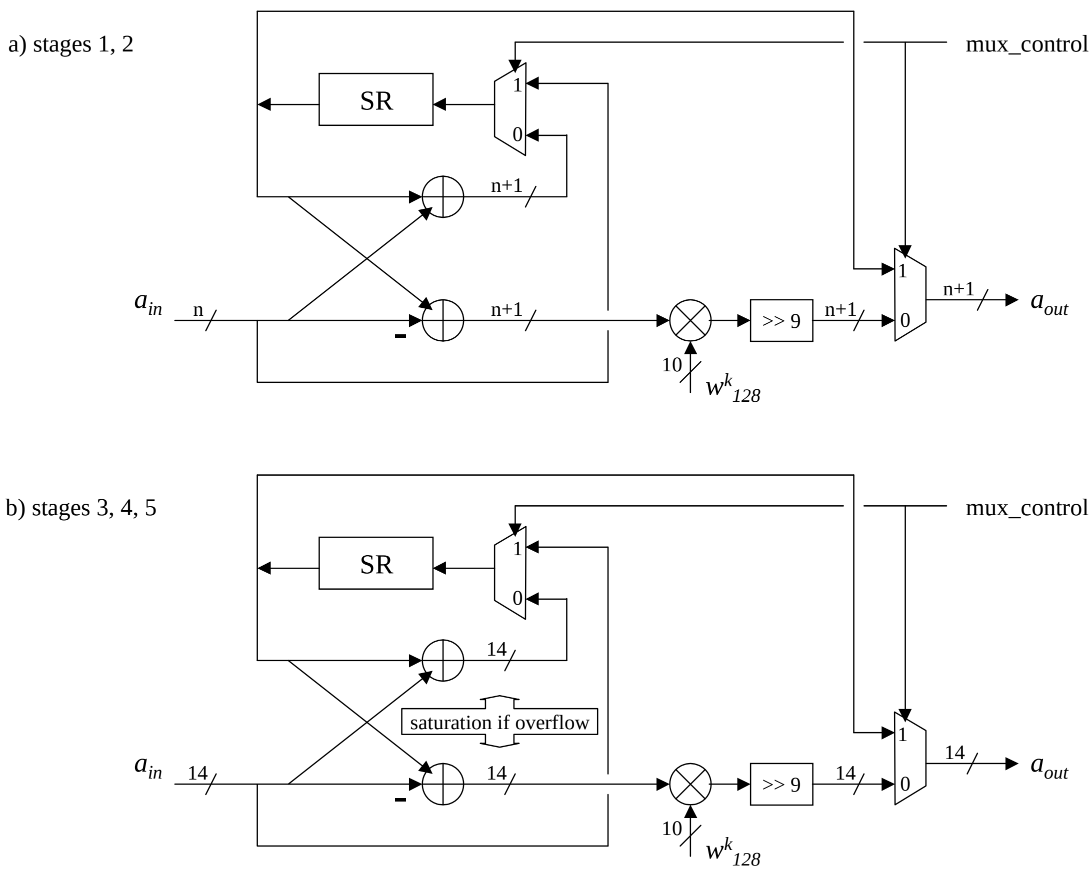
a) Pipelined structure

b) Butterfly structure

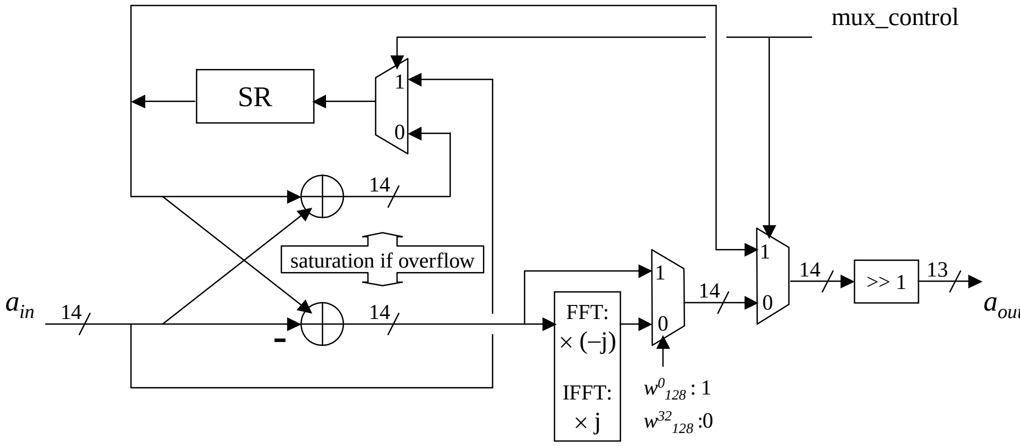
The 128-FFT/IFFT pipeline structure and the general butterfly structure of each stage are shown in Figure 2. All data and arithmetic operations are complex in Figure 2. As indicated in this figure, the 128-FFT structure has the provision to do a 64-point FFT by skipping the first stage and using every other twiddle factors of 128-point FFT. Note that the last two stages have no actual multipliers involved because $W^0_{128} = 1$ and $W^{32}_{128} = -j$ which involves only real-imaginary swapping and sign inversion.

The detailed butterfly structure and bit width of each stage are shown in Figure 3. Generally speaking, an N-point FFT can possibly give $\log_2(N)$ bit growth because each radix-2 butterfly has a maximum growth of one bit to prevent internal overflow. In an OFDM system, the received data have a larger dynamic range before the FFT due to the higher peak-to-average power ratio (PAR) than at its output. After executing the FFT, the data’s dynamic range will be reduced. Computer simulations have shown that with 12 bits input and maintaining 12dB PAR headroom, the receiver FFT can actually saturate the internal stages at 14 bits with negligible consequences.

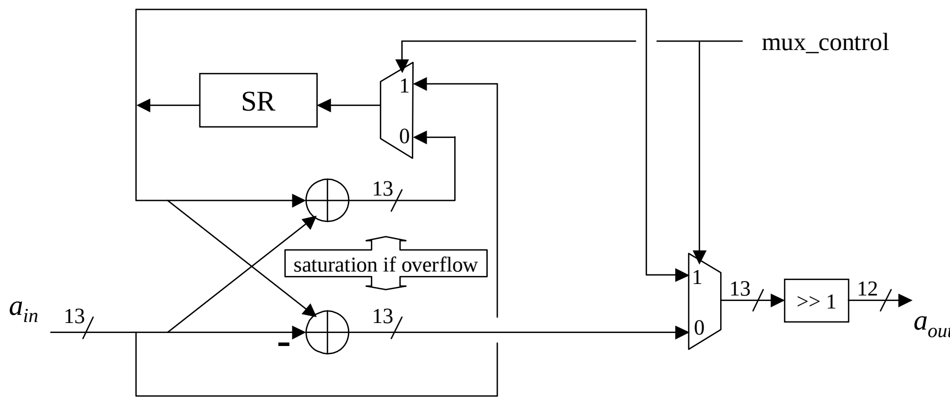
Figure 3: Detailed butterfly structure of each stage in 128-FFT/IFFT



c) stage 6



d) stage 7



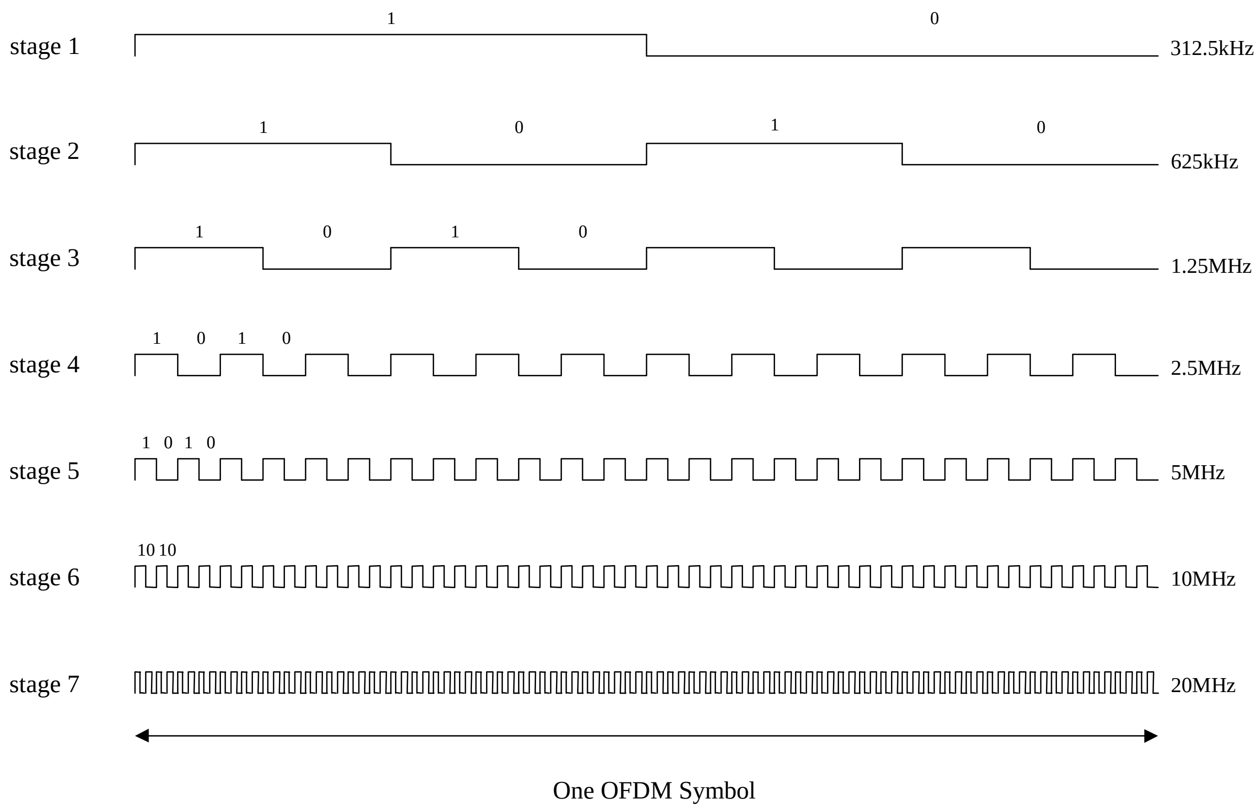
Computer simulations have also shown that the LSB of the last two stages can be neglected. This leads to an acceptable FFT output data bit width of 12 bits, as indicated in Figure 2 and Figure 3.

In the OFDM transmitter, the data's dynamic range will increase after executing the IFFT. In order to use the same FFT circuitry to do the IFFT, we use 9 bits input to the IFFT. With 14-bit internal stages, the saturation can only occur in the last two stages and if present, 15dB PAR clipping results which has negligible degradation to the system performance. As with the FFT, the LSB of the last two stages of the IFFT can be rounded off. This leads to an IFFT output bit width of 12 bits as for the FFT.

4.1.1 MUX Control Signals

The MUX control signals are shown in Figure 4. All signals have to be synchronized, with the start ticks aligned.

Figure 4: MUX Control Signals in 128-FFT/IFFT



4.1.2 Twiddle Factors

The twiddle factors for the 128-point FFT are given by

$$W_{128}^{k} = e^{-j(\frac{\pi}{64})k} = \cos(\frac{\pi}{64}k) - j\sin(\frac{\pi}{64}k) \quad (3)$$

where *k = 0,1, … ,63*. The twiddle factors of for the 128-point IFFT are the conjugate of (3), i.e.

$$W_{128}^{-k} = e^{j(\frac{\pi}{64})k} = \cos(\frac{\pi}{64}k) + j\sin(\frac{\pi}{64}k) \quad (4)$$

A 10-bit (Q9 format) representation for the 128-FFT twiddle factors is provided in Table 9. The twiddle factors of the 128-IFFT can be obtain from the same table by simply negating the imaginary part.

We've noted that the size of Table 1 can be reduced by exploiting the symmetry of the cosine and sine functions involved. Only half of the table needs to be stored. We've also noted that we can only store the unsigned values and save the sign bit. Thus the actual table size can be reduced to 32 x 9 bits.

Table 1 128-FFT twiddles Factors (10-bit)

Index *k*	I	Q	I (binary)	Q (binary)
0	511[1]	0	0111111111	0000000000
1	511	-25	0111111111	1111100111
2	510	-50	0111111110	1111001110
3	506	-75	0111111010	1110110101
4	502	-100	0111110110	1110011100
5	497	-124	0111110001	1110000100
6	490	-149	0111101010	1101101011
7	482	-172	0111100010	1101010100
8	473	-196	0111011001	1100111100
9	463	-219	0111001111	1100100101
10	452	-241	0111000100	1100001111
11	439	-263	0110110111	1011111001
12	426	-284	0110101010	1011100100
13	411	-305	0110011011	1011001111
14	396	-325	0110001100	1010111011
15	379	-344	0101111011	1010101000
16	362	-362	0101101010	1010010110
17	344	-379	0101011000	1010000101
18	325	-396	0101000101	1001110100
19	305	-411	0100110001	1001100101

[1] The actual value is 512. It's saturated to 511 to save one bit width. See section 5 for technical backup.

20	284	-426	0100011100	1001010110
21	263	-439	0100000111	1001001001
22	241	-452	0011110001	1000111100
23	219	-463	0011011011	1000110001
24	196	-473	0011000100	1000100111
25	172	-482	0010101100	1000011110
26	149	-490	0010010101	1000010110
27	124	-497	0001111100	1000001111
28	100	-502	0001100100	1000001010
29	75	-506	0001001011	1000000110
30	50	-510	0000110010	1000000010
31	25	-511	0000011001	1000000001
32	0	-511[2]	0000000000	1000000001
33	-25	-511	1111100111	1000000001
34	-50	-510	1111001110	1000000010
35	-75	-506	1110110101	1000000110
36	-100	-502	1110011100	1000001010
37	-124	-497	1110000100	1000001111
38	-149	-490	1101101011	1000010110
39	-172	-482	1101010100	1000011110
40	-196	-473	1100111100	1000100111
41	-219	-463	1100100101	1000110001
42	-241	-452	1100001111	1000111100
43	-263	-439	1011111001	1001001001
44	-284	-426	1011100100	1001010110
45	-305	-411	1011001111	1001100101
46	-325	-396	1010111011	1001110100
47	-344	-379	1010101000	1010000101
48	-362	-362	1010010110	1010010110
49	-379	-344	1010000101	1010101000
50	-396	-325	1001110100	1010111011
51	-411	-305	1001100101	1011001111
52	-426	-284	1001010110	1011100100
53	-439	-263	1001001001	1011111001
54	-452	-241	1000111100	1100001111
55	-463	-219	1000110001	1100100101
56	-473	-196	1000100111	1100111100
57	-482	-172	1000011110	1101010100
58	-490	-149	1000010110	1101101011
59	-497	-124	1000001111	1110000100
60	-502	-100	1000001010	1110011100
61	-506	-75	1000000110	1110110101
62	-510	-50	1000000010	1111001110
63	-511	-25	1000000001	1111100111

[2] The actual value is -512. It's saturated to -511 to save one bit width. See section 5 for technical backup.

4.1.3 Output Reordering

The output data of the FFT/IFFT will be in inversed bit-reversed order. Properly reordering of the output data back to the normal order must to be done after executing the FFT/IFFT. This requires a buffer of one OFDM symbol length. The relationship between the input data order and output data order is shown in Table 2. The output order can be derived from the input order as follows:

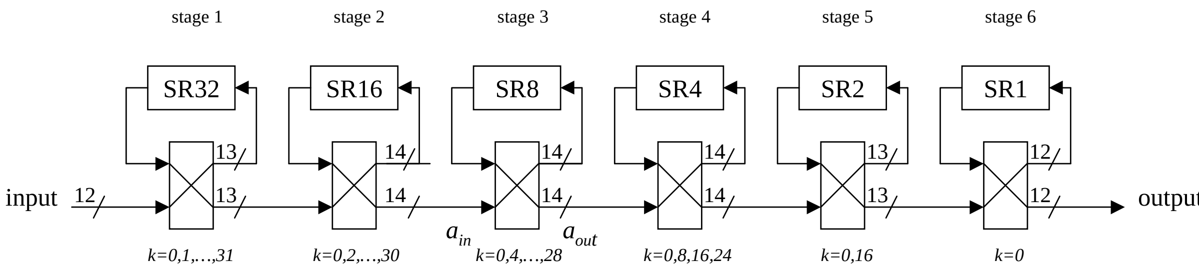
$$out\_order[n:0] = \overline{in\_order[0:n]} \qquad (5)$$

Table 2 Reversed bit-reversed order of 128-FFT/IFFT output

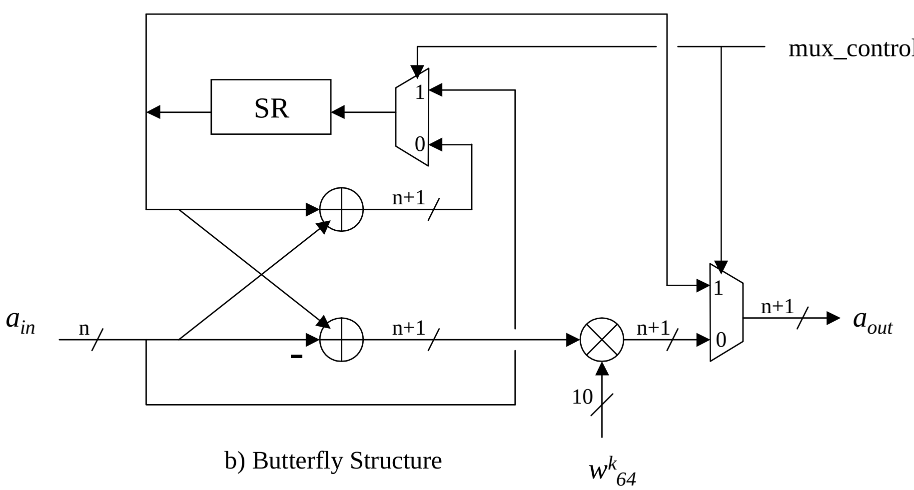
Input Order	Output Order	Input Order	Output Order	Input Order	Output Order	Input Order	Output Order
0	127	32	125	64	126	96	124
1	63	33	61	65	62	97	60
2	95	34	93	66	94	98	92
3	31	35	29	67	30	99	28
4	111	36	109	68	110	100	108
5	47	37	45	69	46	101	44
6	79	38	77	70	78	102	76
7	15	39	13	71	14	103	12
8	119	40	117	72	118	104	116
9	55	41	53	73	54	105	52
10	87	42	85	74	86	106	84
11	23	43	21	75	22	107	20
12	103	44	101	76	102	108	100
13	39	45	37	77	38	109	36
14	71	46	69	78	70	110	68
15	7	47	5	79	6	111	4
16	123	48	121	80	122	112	120
17	59	49	57	81	58	113	56
18	91	50	89	82	90	114	88
19	27	51	25	83	26	115	24
20	107	52	105	84	106	116	104
21	43	53	41	85	42	117	40
22	75	54	73	86	74	118	72
23	11	55	9	87	10	119	8
24	115	56	113	88	114	120	112
25	51	57	49	89	50	121	48
26	83	58	81	90	82	122	80
27	19	59	17	91	18	123	16
28	99	60	97	92	98	124	96
29	35	61	33	93	34	125	32
30	67	62	65	94	66	126	64
31	3	63	1	95	2	127	0

4.2 Block Diagram of Stand Alone 64-point FFT

Figure 5: Stand Alone 64-FFT Implementation Details



a) Pipeline Structure



b) Butterfly Structure

In case we need a stand-alone 64-point FFT, Figure 5 shows the 64-point FFT/IFFT pipeline structure. All discussions of the 128-FFT in the previous sections apply to this 64-FFT also. The twiddle factors of the 64-FFT are given by

$$W_{64}^{k} = e^{-j(\frac{\pi}{32})k} = \cos(\frac{\pi}{32}k) - j\sin(\frac{\pi}{32}k) \qquad (6)$$

where $k = 0,1,\ \ldots\ ,31$. The twiddle factors of the 64-IFFT are the conjugate of (6), i.e.

$$W_{64}^{-k} = e^{j(\frac{\pi}{32})k} = \cos(\frac{\pi}{32}k) + j\sin(\frac{\pi}{32}k) \qquad (7)$$

A 10-bit (Q9 format) representation for the 64-FFT twiddle factors is provided in Table 3. Again, the size of Table 3 can be reduced to 16 x 9 bits as discussed in previous section of 128-FFT.

Note that taking every other entry of Table 1 gives the twiddle factors for the 64-FFT in Table 3. Therefore only Table 1 will be needed if we want both 64 and 128-point FFT.

Table 3 64-FFT twiddles Factors (10-bit)

Index *k*	I	Q	I (binary)	Q (binary)
0	511[3]	0	0111111111	0000000000
1	510	-50	0111111110	1111001110
2	502	-100	0111110110	1110011100
3	490	-149	0111101010	1101101011
4	473	-196	0111011001	1100111100
5	452	-241	0111000100	1100001111
6	426	-284	0110101010	1011100100
7	396	-325	0110001100	1010111011
8	362	-362	0101101010	1010010110
9	325	-396	0101000101	1001110100
10	284	-426	0100011100	1001010110
11	241	-452	0011110001	1000111100
12	196	-473	0011000100	1000100111
13	149	-490	0010010101	1000010110
14	100	-502	0001100100	1000001010
15	50	-510	0000110010	1000000010
16	0	-511[4]	0000000000	1000000001
17	-50	-510	1111001110	1000000010
18	-100	-502	1110011100	1000001010
19	-149	-490	1101101011	1000010110
20	-196	-473	1100111100	1000100111
21	-241	-452	1100001111	1000111100
22	-284	-426	1011100100	1001010110
23	-325	-396	1010111011	1001110100
24	-362	-362	1010010110	1010010110
25	-396	-325	1001110100	1010111011
26	-426	-284	1001010110	1011100100
27	-452	-241	1000111100	1100001111
28	-473	-196	1000100111	1100111100
29	-490	-149	1000010110	1101101011
30	-502	-100	1000001010	1110011100
31	-510	-50	1000000010	1111001110

[3] The actual value is 512. It's saturated to 511 to save one bit width. See section 5 for technical backup.
[4] The actual value is -512. It's saturated to -511 to save one bit width. See section 5 for technical backup.

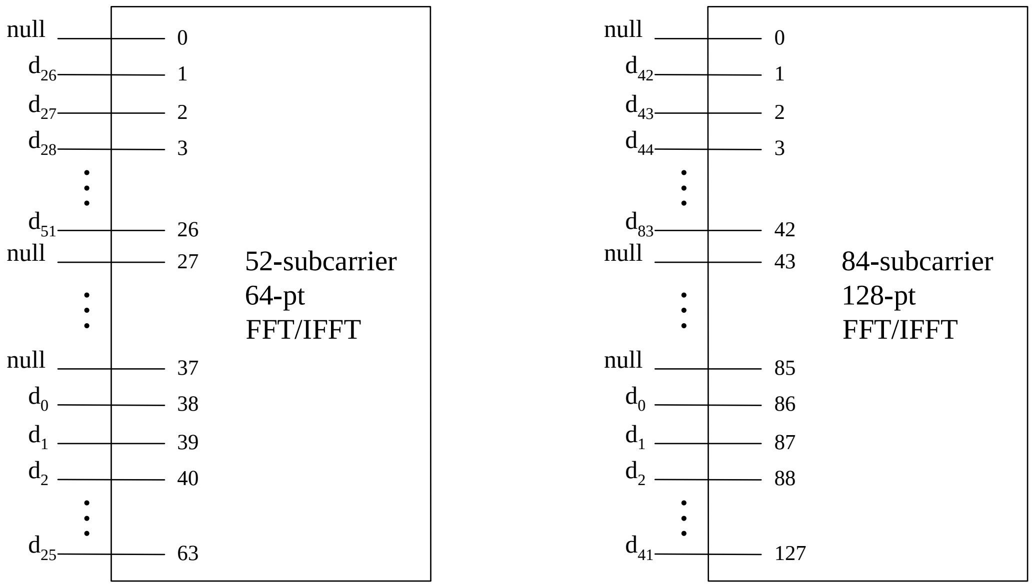
The inversed bit-reversed order of 64-FFT/IFFT output is shown in Table 4.

Table 4 Reversed bit-reversed order of 64-FFT/IFFT output

Input Data Order	Output Data Order	Input Data Order	Output Data Order
0	63	32	62
1	31	33	30
2	47	34	46
3	15	35	14
4	55	36	54
5	23	37	22
6	39	38	38
7	7	39	6
8	59	40	58
9	27	41	26
10	43	42	42
11	11	43	10
12	51	44	50
13	19	45	18
14	35	46	34
15	3	47	2
16	61	48	60
17	29	49	28
18	45	50	44
19	13	51	12
20	53	52	52
21	21	53	20
22	37	54	36
23	5	55	4
24	57	56	56
25	25	57	24
26	41	58	40
27	9	59	8
28	49	60	48
29	17	61	16
30	33	62	32
31	1	63	0

4.3 Subcarriers to FFT bins Mapping

Figure 6: Subcarriers to FFT bins Mapping



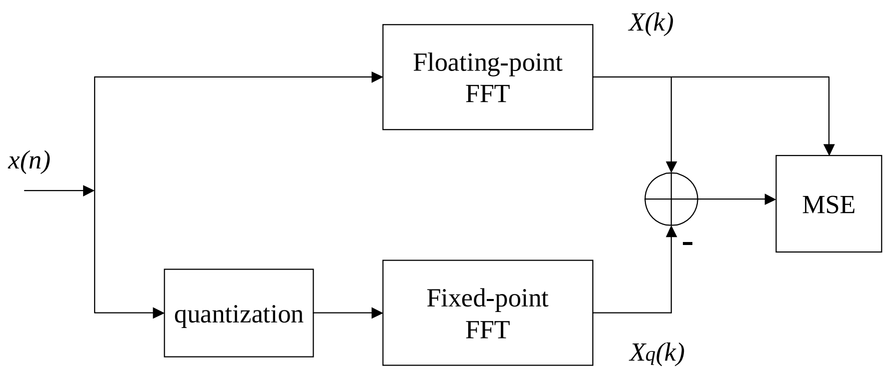
$d_0, d_1, d_2, \ldots, d_{51}$ are 48 data symbols plus 4 pilot symbols.

$d_0, d_1, d_2, \ldots, d_{83}$ are 80 data symbols plus 4 pilot symbols.

At the transmitter, before performing the IFFT, the logical subcarrier indices for the data (and the pilot) symbols need to be mapped into the FFT bin index. This mapping is illustrated in Figure 6. At the receiver the reverse process shall be performed after the FFT.

5 Technical Backup

Figure 7: Simulation of Quantization Effects in FFT



5.1 Quantization Effect of MSE

The mean squared error (MSE) measurement is shown in Figure 7. The MSE is defined as

$$MSE_{dB} = 10\log_{10}\left(\frac{E\{|X(k) - X_q(k)|^2\}}{E\{|X(k)|^2\}}\right) \quad (8)$$

The zero valued subcarriers are excluded from the measurement. Table 5 and Table 6 give the results of 128-pt FFT and IFFT with 84 data subcarriers modulated by 64-QAM signals.

Table 5 MSE of 128-FFT

Input Data Bit Width	Twiddle Factor Bit Width	Output Data Bit Width	MSE (dB)
12 bits[5]	10 bits (with ±512)	14 bits	– 60
		12 bits (round off 2 LSB)	– 59
	10 bits (±512 → ±511)		– 55
Same bit widths, but using truncation instead of rounding cross FFT			– 53

Table 6 MSE of 128-IFFT

Input Data Bit Width	Twiddle Factor Bit Width	Output Data Bit Width	MSE (dB)
9 bits	10 bits (with ±512)	14 bits[6]	– 50
		12 bits (round off 2 LSB)	– 48
	10 bits (±512 → ±511)		– 48
Same bit widths, but using truncation instead of rounding cross FFT			– 45

Table 7 and Table 8 give the results of 64-pt FFT and IFFT with 52 data subcarriers modulated by 64-QAM signals. Note that the MSE of the 64-point FFT is worse than the MSE of the 128-point FFT because the truncation of the LSB in the last two stages has more effect on 64-point FFT than the 128-point FFT.

Table 7 MSE of 64-FFT

Input Data Bit Width	Twiddle Factor Bit Width	Output Data Bit Width	MSE (dB)
12 bits	10 bits	12 bits	– 53

Note: same configuration as 128-FFT

Table 8 MSE of 64-IFFT

Input Data Bit Width	Twiddle Factor Bit Width	Output Data Bit Width	MSE (dB)
9 bits	10 bits	12 bits	– 43

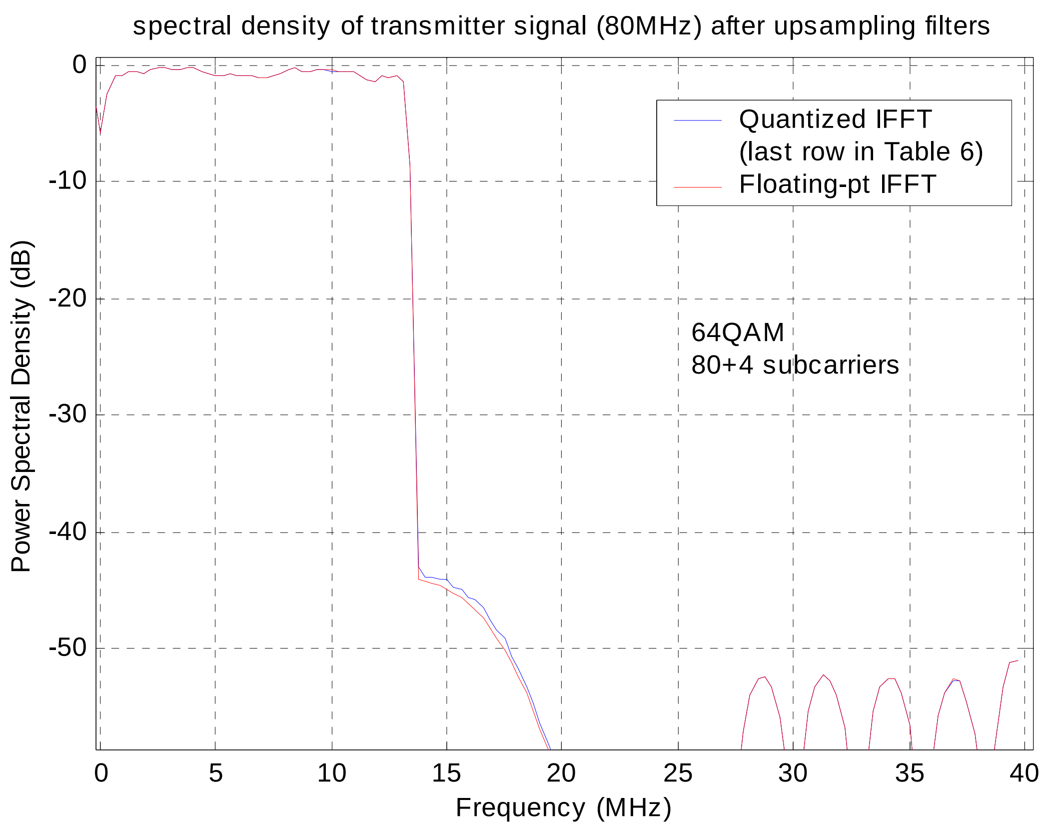
Note: same configuration as 128-IFFT

[5] Q9 format. Including 12dB PAR, but not including noise and fading.
[6] With a saturation at 15dB PAR.

5.2 Quantization Effect of Out-of-Band Radiation

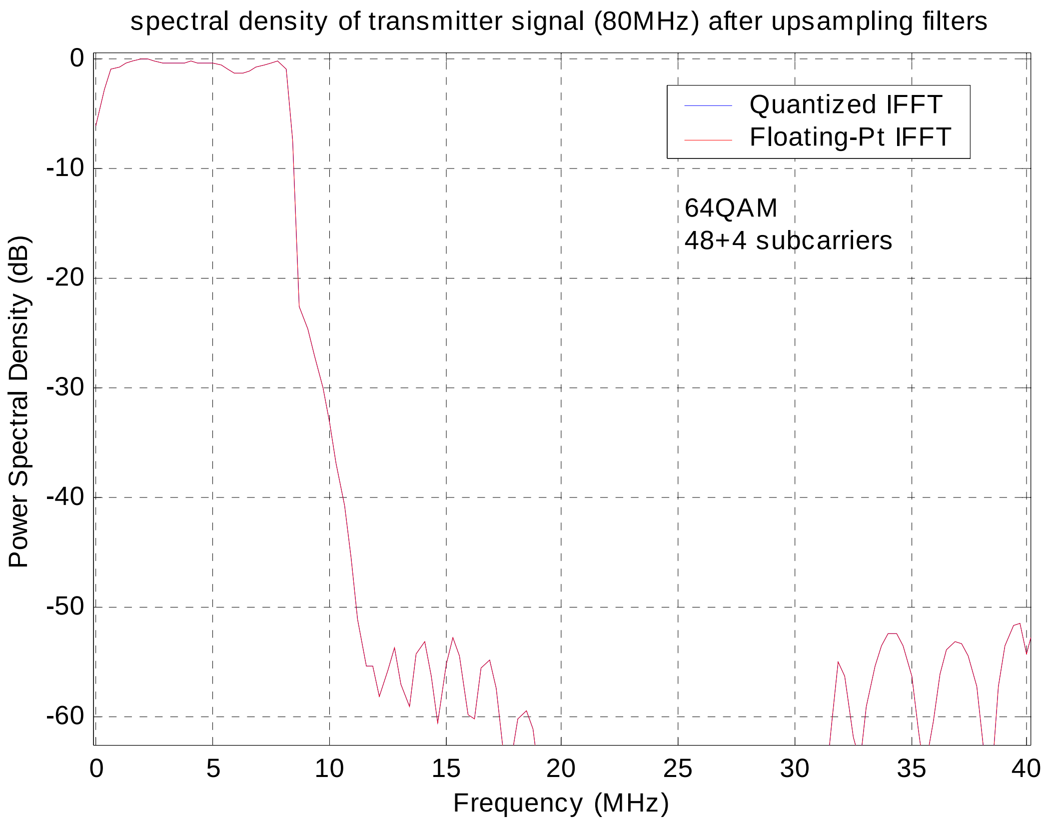
The quantization and saturation in the IFFT processing will cause out-of-band radiation in the transmitter signal power spectral density. The up-sampling filters following the IFFT will suppress the out-of-band radiation to an acceptable level. Figure 8 and Figure 9 give the spectral density of transmitter signal (80MHz) after the up-sampling filters[7].

Figure 8: 64-point IFFT in 802.11a mode



[7] Taken from Magis Docts E10016 and E10017.

Figure 9: 64-point IFFT in 802.11a mode



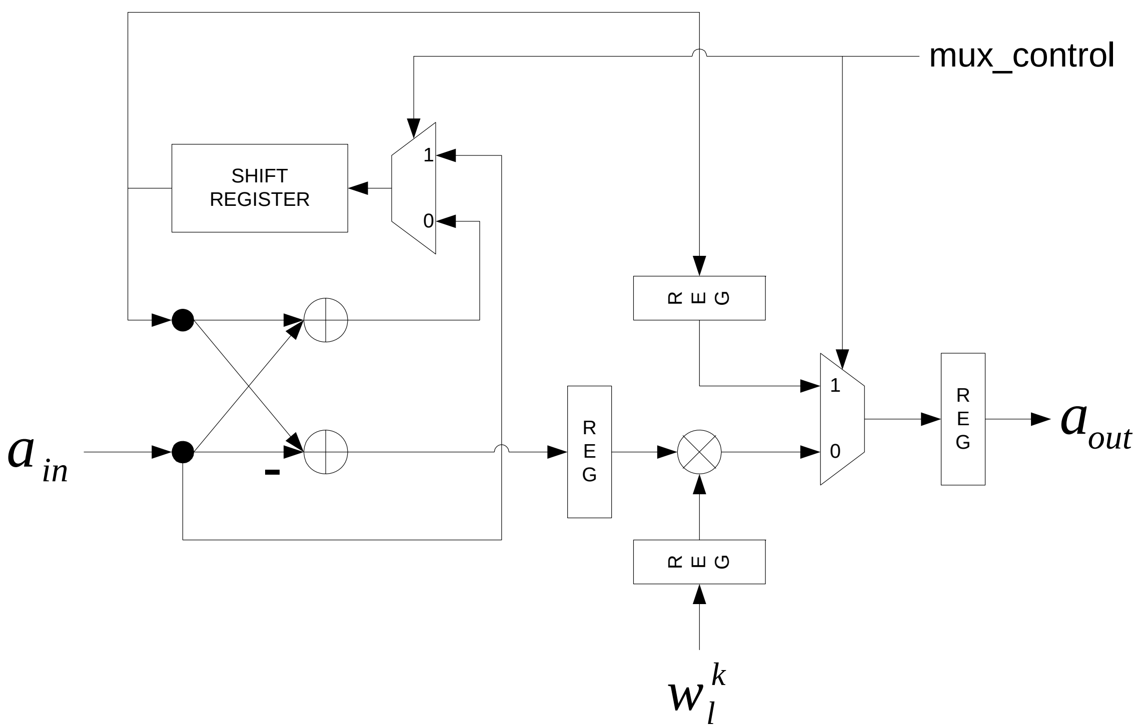
6 Pisces Implementation Details

This section describes the variations to the basic IFFT/FFT design for the Pisces development effort. The variations were primarily driven by two issues:

The hardware must operate at 40MHz.
Minimize development time and risk.

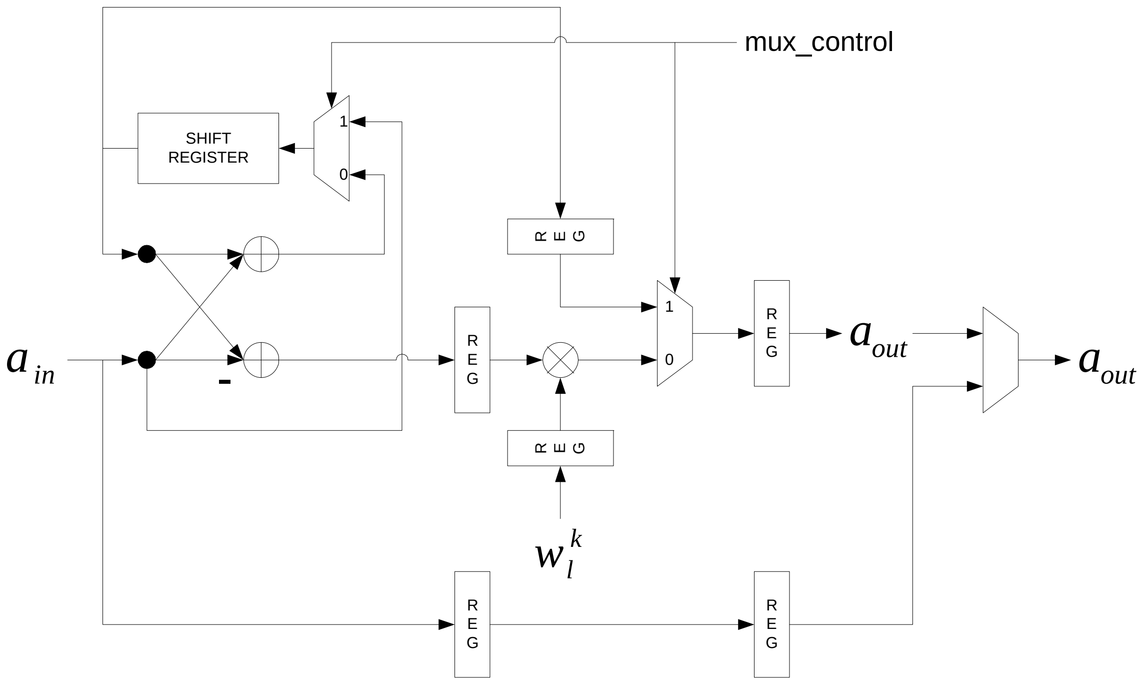
The multipliers in the FFT design were taking up to 20ns of delay for a 17x10 unsigned multiplication. To increase the speed of the design, pipeline delays were added before and after the multipliers. This increased the throughput delay, but allowed the design to operate at the required 40MHz rate. Stages 2-5 of the FFT have internal pipeline registers as shown in the figure below. Stages 6,7 do not require multipliers and do not require the pipeline registers.

Figure 10: Pisces FFT Stages 2-5



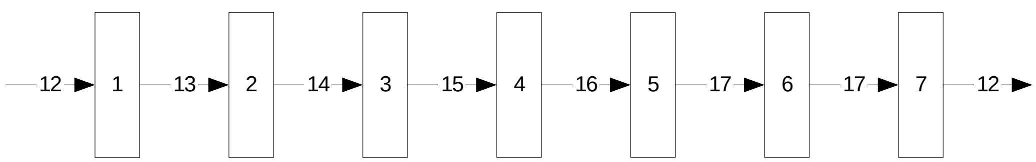
The first stage of the FFT has an additional path which bypasses the first stage to support a 64 point FFT. There are two register delays inserted in this path in order to keep the throughput delay consistent with the 128-point FFT delay.

Figure 11: Pisces FFT Stage 1



To minimize the design time and risk for the POC and Orion 1 efforts, the saturation logic within each stage of the FFT was not implemented. Instead, each stage of the FFT was grown at the output by one bit in order to avoid any overflow issues from the butterfly additions. This costs more gates, but they come for "free" in the POC fpga platform. Multiplies are done as unsigned multiplies without bit growth in the result. This was done by rounding off the LSBs.

Figure 12: Pisces FFT bit widths



As seen in Figure 12, each stage of the FFT grows one bit up to stage 6. The output of stage 6 is 18 bits. The LSB is truncated resulting in 17 bits to drive stage 7. Stage 7's output is also 18 bits. The LSB is truncated and the 5 MSBs are saturated resulting in a 12 bit output.

The pipeline structure of the FFT requires that one extra OFDM symbol (data_valid) be driven into its input in order to flush the last OFDM symbol out of the FFT. This is handled by generating an internal data valid which is one OFDM symbol delay from the input data valid. The or function of the external data valid and internal data valid is used to drive the FFT state-machine.

Figure 13: FFT input/output timing

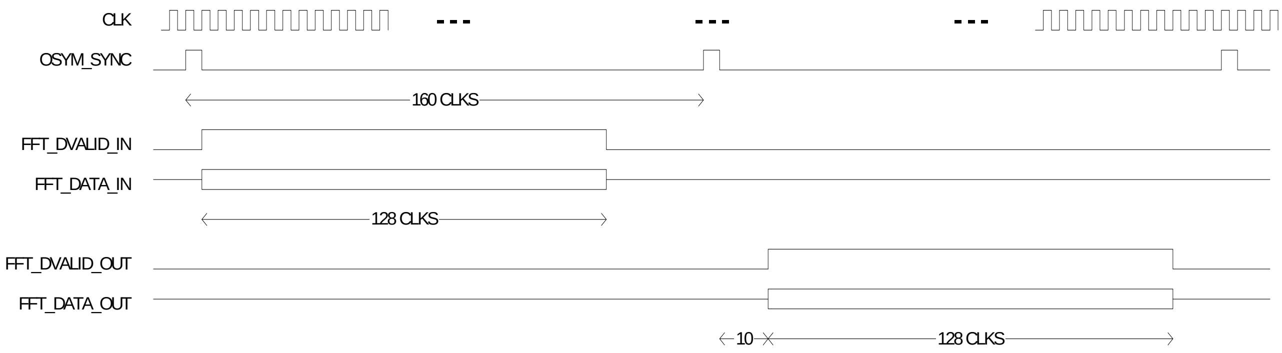


Figure 13 shows an example timing diagram for a 128 point FFT IO. On the FFT input, FFT data valid and data can occur anywhere within the window marked by the OFDM symbol syncs. The data must not cross an OFDM symbol sync boundary or unexpected results will occur. On the FFT output, the FFT data is generated one OFDM symbol after the input data and has a fixed 10 clock delay from the falling edge of the OFDM symbol sync. In the case of a 64 point FFT, all logic operates with the 40MHz enabled by a 20MHz enable. There are 64 clocks of input/output data. The delay from OFDM symbol sync to FFT data output is ten 20MHz clocks. Note that at present, when the 64 point FFT is used, the power for the first stage of the FFT block is still on, which costs about half of the entire FFT memory (shift register), even through data bypasses it.

7 System Design Close-Out

7.1 Data Flow In/Out of Module

See 2.1.

7.2 Module Control

See 2.3.

7.2.1.1 Programmer's or User's Manual

7.3 Other Issues

Since the FFT is one of the most computationally intensive modules in the system, care must be taken in the implementation to minimize the cost. For the PoC and Orion 1, three identical FFT blocks have been implemented (one for the transmit and two for the receive) and take about 20% of a XCV2000. In the final product, sharing one FFT for both the transmitter and the receiver will save one FFT block. In other words, one FFT for both transmit and receive and the other FFT for receive only.

In addition, one area of improvement includes saturating the output of each stage in the FFT to a fixed bit-width. For example, the implementation in Figure 12 can be reduced to that in Figure 1.

The other improvement is to minimize the bit-width for the receive FFT. The reason is that the transmitter and the receiver have different precision requirement for the FFT. For the receiver, the dynamic range required is determined by the equalizer or limited by the ADC, which is 10 bits. Further improvement may be possible, based on the refined design of the surrounding blocks.

Each stage butterfly structure requires its own twiddle factor table, which can be optimized reducing the table memory by exploiting the symmetry of the sine and cosine functions to reduce the actual table size.

7.4 802.11a Mode Coverage Compliance

The FFT/IFFT module has been successfully employed for receive in the fixed point C simulations for all modes in the table below. The ASIC implementation has been demonstrated to work for all modes but BPSK rate ¾, QPSK rate ¾, 16 QAM rate ½, and 64 QAM rate ½. These modes have not been tested in the lab.

Table 9 Phy-Mode Compliance Table for Digital RSSI Module

QAM Size	Code Rate	Throughput Rate, Mbps	AP Operation Verified	RT Operation Verified	Special Issues / Caveats
BPSK	½	6	Y	Y	
BPSK	¾	9	Y	Y	
QPSK	½	12	Y	Y	
QPSK	¾	18	Y	Y	
16QAM	½	24	Y	Y	
16QAM	¾	36	Y	Y	
64QAM	¾	54	Y	Y	
64QAM	½	36	Y	Y	Non 802.11a mode, but desirable if achievable.

7.5 Message Coverage Compliance

Table 10 Message Coverage Compliance Table for FFT/IFFT RSSI Module

Message Designator	Mssg Description	AP Operation Verified	RT Operation Verified	Special Issues / Caveats
apf0	Simplex downlink data			
apf1	Simplex downlink data with duplex control			
apf2	Simplex uplink data			
apf3	Simplex uplink data with duplex control			
apf4	Duplex data with duplex control			
rtf0	Simplex downlink data			
rtf1	Simplex dowlink data with duplex control			
rtf2	Simplex uplink data			
rtf2~	Simplex uplink data			
rtf3	Simplex uplink data with duplex control			
rtf4	Duplex data with duplex control			
rtf5	Duplex data, duplex control, broadcast data			

8 Test Vectors of 128-FFT/IFFT

The FFT module level testing and bit matching was done in a testbench. A description of the testbench file can be found in the Orion VSS database: $/Orion/phy_top/phy_tx/fft/readme.txt.

8.1 Test Vectors for 128-FFT:

One OFDM symbol with 84 subcarriers and 64QAM modulation is provided in Table 11.

Table 11 Test Vector for 128-FFT

Index	Input 12 bits in normal order		Output 12 bits			
			before reordering		after reordering	
	I	Q	I	Q	I	Q
0	-603	-86	-1377	-824	2	-1
1	-161	84	1	2	-834	-835
2	-20	-10	275	-1375	-279	825
3	-728	-746	275	280	-832	-1381
4	-435	-836	1382	-274	-1382	-278
5	518	234	3	1	-1933	-829
6	388	671	0	4	-830	1376
7	38	83	-1378	-827	-275	-828
8	213	-114	1381	277	-1929	-1382
9	-47	-77	0	3	1927	273
10	-498	-478	-826	-827	-827	-832
11	-371	-473	-828	-827	-1381	-1926
12	30	68	-1930	-1376	-1381	273
13	492	28	-275	1930	-828	-1378
14	752	-340	0	1	272	825
15	326	-51	-275	-828	-1378	-827
16	-313	329	-828	274	274	274
17	-453	32	-2	-2	-1930	1928
18	-138	-299	824	-1381	1926	1375
19	211	-152	824	1933	1928	274
20	44	74	-829	-823	276	-829
21	-479	333	-1	0	1926	-827
22	-512	665	-1	4	278	-278
23	-272	455	-1381	-1926	-828	-827
24	-333	-249	-275	-1929	1924	1375
25	-171	-349	1	3	1927	827
26	242	287	-1	3	1926	-1382
27	297	501	1928	274	824	1933
28	234	-46	830	1931	1381	-1382
29	77	-354	275	-1935	276	830
30	-411	76	-1	-1	279	827
31	-473	420	-832	-1381	275	280

32	-69	69	-272	-1372	-1931	-828
33	-104	-317	1	4	277	277
34	-177	-229	-1378	-1924	-277	-275
35	148	-166	276	830	275	-1935
36	193	-243	825	-1379	-828	-828
37	89	-41	0	3	-273	-827
38	112	144	1	1	-825	277
39	-171	-80	-828	-1378	-275	1930
40	-126	-292	1929	-1378	-1930	1380
41	532	-92	-3	3	-1931	-275
42	434	486	1	3	-1376	-276
43	-169	1013	1926	-827	-1	0
44	135	715	-276	-276	0	-3
45	505	-112	-273	-827	0	3
46	148	-234	-3	-1	-1	-5
47	21	36	-1933	-829	3	1
48	4	-246	-1928	-826	-1	-2
49	-317	-307	1	2	1	-1
50	6	142	-276	-1929	4	1
51	578	-69	1927	827	1	3
52	129	-525	1377	1929	0	1
53	-310	-199	-1931	-275	-3	3
54	242	-68	-2	-2	2	0
55	472	-407	1927	273	0	3
56	3	68	274	1927	-1	-3
57	80	671	1	-1	1	2
58	457	151	0	-2	1	-6
59	183	-250	-1930	1928	-2	-2
60	-59	147	-1930	828	0	1
61	191	59	277	277	1	4
62	-58	-218	0	-2	-1	-1
63	-626	394	-834	-835	1	2
64	-224	845	277	-1378	0	0
65	459	472	-1	-1	0	-2
66	103	430	-1926	-278	0	1
67	-336	542	279	827	-1	-1
68	70	-173	1929	1376	1	-1
69	355	-619	-1	-5	-3	-1
70	247	63	-2	-2	-1	-3
71	431	349	272	825	0	1
72	522	-400	277	826	3	0
73	135	-645	2	0	-2	-2
74	-211	144	-1378	-828	3	0
75	-437	763	278	-278	-1	4
76	-649	531	-1930	-275	2	-1
77	-490	-51	-825	277	1	1
78	8	-271	-1	-3	-2	-2

79	468	-176	-830	1376	0	4
80	589	-329	-828	1928	-1	1
81	92	-440	1	-6	0	-2
82	-483	40	824	1931	-1	-2
83	-223	298	1926	-1382	-1	3
84	405	-198	-1928	1378	0	0
85	405	-363	-1376	-276	1	3
86	-20	69	3	0	-1378	-828
87	-161	193	-827	-832	-826	-827
88	97	117	-1929	-1378	1925	274
89	277	100	4	1	-276	-1929
90	-23	-207	-1	-2	824	1931
91	-345	-326	1926	1375	824	-1381
92	-250	-24	-1378	-829	-1931	1928
93	-271	-92	-277	-275	-1378	-1924
94	-509	-173	0	1	-1926	-278
95	-292	254	-279	825	275	-1375
96	207	69	-1924	823	1930	827
97	275	-638	0	1	-1930	828
98	-144	-311	-1931	1928	-1378	-829
99	-471	469	1381	-1382	830	1931
100	-11	372	1931	1929	829	1379
101	646	5	0	-3	-276	-276
102	173	-8	2	-1	-1930	-275
103	-491	165	-1381	273	-1930	-1376
104	81	461	-1927	275	-825	276
105	459	329	0	1	1377	1929
106	-68	-234	0	0	-1928	1378
107	56	-248	276	-829	-829	-823
108	237	-125	829	1379	1931	1929
109	-543	-425	-828	-828	825	-1379
110	-510	27	1	-1	1929	1376
111	390	822	-1382	-278	1382	-274
112	272	315	-276	826	275	828
113	117	-197	-1	-3	274	1927
114	798	431	1925	274	-1929	-1378
115	664	265	1924	1375	-275	-1929
116	19	-816	-825	276	-1927	275
117	182	-735	-1930	1380	1929	-1378
118	208	25	3	0	277	826
119	-154	116	-1929	-1382	1381	277
120	-43	-5	275	828	-276	826
121	47	-166	-1	-2	-1928	-826
122	0	-344	-1	1	-828	1928
123	118	119	274	274	-828	274
124	-367	527	1930	827	-1924	823
125	-930	11	-1931	-828	-272	-1372

126	-532	-428	0	0	277	-1378
127	-334	-253	2	-1	-1377	-824

8.2 Test Vectors for 128-IFFT:

One OFDM symbol with 84 subcarriers and 64QAM modulation is provided in Table 12.

Table 12 Test Vector for 128-IFFT

Index	Input 9 bits in normal order		Output 12 bits			
			before reordering		after reordering	
	I	Q	I	Q	I	Q
0	0	0	-236	308	-159	-50
1	122	73	390	-480	49	-372
2	-122	-73	130	-351	191	54
3	-171	73	-77	-167	304	8
4	-171	-24	-36	348	-39	-534
5	24	24	-22	-148	-445	-176
6	73	-171	42	10	-88	339
7	122	24	-30	-391	188	11
8	-24	24	-11	-125	-124	-12
9	-73	-122	-273	273	8	384
10	-122	-122	-108	-440	272	84
11	-171	24	294	-35	-206	-327
12	73	-73	325	-286	-413	-163
13	-73	-171	-483	71	36	-125
14	24	171	-14	48	108	-355
15	122	24	188	11	-30	-391
16	171	-171	-154	34	8	-212
17	-122	122	-63	-18	-203	66
18	-73	73	110	140	-107	152
19	24	-171	-82	369	494	39
20	122	-171	135	-150	364	212
21	122	122	375	60	-242	320
22	-171	73	139	469	-75	-34
23	73	-24	-206	-327	294	-35
24	-122	171	22	-348	170	277
25	-171	-73	445	79	22	2
26	24	24	177	-248	-81	-174
27	24	122	494	39	-82	369
28	-24	-122	-127	-102	242	541
29	-171	-24	-91	624	230	106
30	-171	-24	-27	-67	-188	-89
31	-73	-171	304	8	-77	-167

32	171	24	-265	-85	244	-330
33	73	122	-176	30	31	-136
34	-122	-122	-297	139	-173	338
35	-24	-24	230	106	-91	624
36	-73	24	454	-161	-149	473
37	-122	171	-252	237	-277	50
38	-122	73	105	-512	-400	-85
39	-73	-171	36	-125	-483	71
40	24	-73	-89	92	-172	12
41	122	-171	1	-400	312	-41
42	-122	122	-52	211	466	90
43	0	0	-242	320	375	60
44	0	0	-221	-27	67	96
45	0	0	-277	50	-252	237
46	0	0	-261	-5	-97	-28
47	0	0	-445	-176	-22	-148
48	0	0	120	227	-411	277
49	0	0	-75	77	-222	291
50	0	0	269	-143	475	-47
51	0	0	22	2	445	79
52	0	0	-120	117	65	7
53	0	0	312	-41	1	-400
54	0	0	378	-71	-206	-136
55	0	0	8	384	-273	273
56	0	0	400	418	51	81
57	0	0	-222	291	-75	77
58	0	0	-261	166	-358	277
59	0	0	-203	66	-63	-18
60	0	0	-90	251	5	-214
61	0	0	31	-136	-176	30
62	0	0	-438	-171	254	-60
63	0	0	49	-372	390	-480
64	0	0	-59	37	-305	-537
65	0	0	254	-60	-438	-171
66	0	0	-243	107	31	110
67	0	0	-188	-89	-27	-67
68	0	0	282	360	-255	-296
69	0	0	-97	-28	-261	-5
70	0	0	-96	-473	-338	301
71	0	0	108	-355	-14	48
72	0	0	-379	76	580	-227
73	0	0	-206	-136	378	-71
74	0	0	-406	2	-114	281
75	0	0	-75	-34	139	469
76	0	0	-213	-553	419	98
77	0	0	-400	-85	105	-512
78	0	0	-338	301	-96	-473

79	0	0	-88	339	42	10
80	0	0	205	147	-9	186
81	0	0	-358	277	-261	166
82	0	0	191	143	-237	31
83	0	0	-81	-174	177	-248
84	0	0	-9	95	372	-132
85	0	0	466	90	-52	211
86	24	171	-114	281	-406	2
87	171	171	272	84	-108	-440
88	24	24	-37	74	265	-449
89	171	73	475	-47	269	-143
90	-24	-122	-237	31	191	143
91	122	122	-107	152	110	140
92	73	-73	-532	88	-102	-1
93	73	-122	-173	338	-297	139
94	-24	-73	31	110	-243	107
95	-122	-24	191	54	130	-351
96	73	-171	-540	37	366	-257
97	-24	73	5	-214	-90	251
98	-122	-171	-102	-1	-532	88
99	-122	171	242	541	-127	-102
100	24	-73	221	-496	188	208
101	171	24	67	96	-221	-27
102	-171	-73	419	98	-213	-553
103	171	-73	-413	-163	325	-286
104	-171	-122	323	-234	250	128
105	-24	73	65	7	-120	117
106	-73	122	372	-132	-9	95
107	-73	-24	364	212	135	-150
108	73	-171	188	208	221	-496
109	-24	122	-149	473	454	-161
110	171	122	-255	-296	282	360
111	122	24	-39	-534	-36	348
112	-122	122	142	-9	263	329
113	73	122	51	81	400	418
114	-73	122	265	-449	-37	74
115	24	171	170	277	22	-348
116	171	-171	250	128	323	-234
117	-73	73	-172	12	-89	92
118	122	-122	580	-227	-379	76
119	73	73	-124	-12	-11	-125
120	-24	-122	263	329	142	-9
121	-24	122	-411	277	120	227
122	-24	73	-9	186	205	147
123	73	73	8	-212	-154	34
124	-122	-24	366	-257	-540	37
125	24	-24	244	-330	-265	-85

126	24	-73	-305	-537	-59	37
127	-73	-24	-159	-50	-236	308

8.3 Test Vectors for 64-FFT:

One OFDM symbol with 52 subcarriers and 64QAM modulation is provided in Table 13.

Table 13 Test Vector for 64-FFT

Index	Input 12 bits in normal order		Output 12 bits			
			before reordering		after reordering	
	I	Q	I	Q	I	Q
0	131	-88	-525	524	-1	1
1	-648	-310	-1	-4	-531	-530
2	-710	153	875	-1230	525	-1229
3	-436	-677	171	1224	-528	-880
4	211	-461	-1226	-878	-527	-177
5	622	634	176	1226	-175	875
6	348	321	177	-178	-881	172
7	354	182	-1228	874	-1228	874
8	59	-95	-178	874	524	-526
9	-77	48	-1	-2	-177	1224
10	211	512	174	-1228	-175	875
11	371	502	1224	1226	1224	1226
12	419	255	874	1226	174	1224
13	1	-231	1224	-177	-175	877
14	-34	261	-2	-1	1226	-1224
15	85	180	-528	-880	171	1224
16	153	22	-1225	-877	-527	-527
17	430	354	2	1	-177	-878
18	237	-442	-874	-876	174	-876
19	222	-169	-175	877	1224	-177
20	-11	-309	1228	875	-877	876
21	-329	-921	-1229	-877	-1229	-877
22	-425	-452	-2	-2	175	-1227
23	-692	-684	-175	875	176	1226
24	3	313	1227	-875	176	175
25	263	754	-879	-174	-879	-174
26	-120	-270	1225	1225	175	1226
27	19	77	-177	1224	-1	-2
28	-202	139	876	-177	-3	-2
29	-21	343	-177	-878	2	1
30	190	430	1	-1	-2	-2
31	-21	-417	-531	-530	-1	-4
32	88	-394	-1226	1227	0	0
33	329	-62	-2	-2	1	-1

34	464	201	523	-1224	-1	1
35	81	347	1226	-1224	-2	-1
36	137	-37	1224	175	0	1
37	118	-292	175	-1227	-2	-2
38	-222	454	1225	877	1225	877
39	198	414	-881	172	177	-178
40	28	-869	174	-1226	-176	-525
41	258	-19	175	1226	1225	1225
42	420	527	-1226	174	-1226	174
43	-520	-134	-175	875	174	-1228
44	-184	702	-876	-875	-176	-177
45	-38	-32	174	-876	-874	-876
46	-49	-907	-1	1	523	-1224
47	586	325	525	-1229	875	-1230
48	-197	285	1225	174	876	-526
49	-438	-586	-3	-2	876	-177
50	282	-31	-176	-177	-876	-875
51	-13	837	174	1224	874	1226
52	-75	369	527	-176	527	-176
53	367	-119	-877	876	1228	875
54	-150	-173	0	1	1224	175
55	-600	-377	-527	-177	-1226	-878
56	-90	-400	-526	-525	-526	-525
57	-365	-632	176	175	1227	-875
58	-609	227	-176	-525	174	-1226
59	28	663	524	-526	-178	874
60	-120	-483	876	-526	1225	174
61	-472	-331	-527	-527	-1225	-877
62	-184	242	0	0	-1226	1227
63	336	335	-1	1	-525	524

8.4 Test Vectors for 64-IFFT:

One OFDM symbol with 52 subcarriers and 64QAM modulation is provided in Table 14.

Table 14 Test Vector for 64-IFFT

Index	Input 9 bits in normal order		Output 12 bits			
			before reordering		after reordering	
	I	Q	I	Q	I	Q
0	0	0	-91	-78	361	-296
1	-110	66	92	109	72	173
2	66	110	350	-395	-208	212
3	-66	-153	-248	-304	-249	-264
4	110	153	43	71	-157	-11
5	-110	110	-43	-25	24	127
6	-22	-22	-63	94	-61	240
7	66	-110	94	3	94	3
8	66	-153	242	58	180	-196
9	-66	22	93	-61	-26	61
10	110	153	103	-5	76	-444
11	153	-110	65	-321	65	-321
12	153	-22	-159	37	-42	297
13	110	-66	96	97	-53	130
14	-110	66	185	-66	-238	-15
15	153	-153	-249	-264	-248	-304
16	153	-22	147	35	-220	-242
17	153	-66	-1	81	-242	35
18	110	-66	-103	119	76	-58
19	-153	-153	-53	130	96	97
20	66	66	44	230	-269	24
21	-66	-66	-121	-70	-121	-70
22	22	-110	-202	242	57	119
23	153	153	24	127	-43	-25
24	-22	-66	-18	-94	126	-416
25	22	-110	121	-280	121	-280
26	-66	-22	-158	-215	48	214
27	0	0	-26	61	93	-61
28	0	0	-79	375	-138	-152
29	0	0	-242	35	-1	81
30	0	0	65	92	233	-201
31	0	0	72	173	92	109
32	0	0	-203	-78	165	339
33	0	0	233	-201	65	92
34	0	0	91	-31	-15	24
35	0	0	-238	-15	185	-66

36	0	0	-216	118	-121	332
37	0	0	57	119	-202	242
38	110	-22	164	-244	164	-244
39	22	-153	-61	240	-63	94
40	153	-66	119	201	-269	-24
41	-66	-153	48	214	-158	-215
42	153	110	-216	202	-216	202
43	-22	110	76	-444	103	-5
44	-110	-22	113	426	295	-214
45	153	-153	76	-58	-103	119
46	22	22	-15	24	91	-31
47	-22	-22	-208	212	350	-395
48	-110	-153	144	111	-89	-154
49	110	22	-138	-152	-79	375
50	153	153	295	-214	113	426
51	-22	-66	-42	297	-159	37
52	-110	110	-70	49	-70	49
53	-66	22	-269	24	44	230
54	110	66	-121	332	-216	118
55	22	110	-157	-11	43	71
56	22	-110	266	-153	266	-153
57	66	-110	126	-416	-18	-94
58	66	22	-269	-24	119	201
59	-110	-153	180	-196	242	58
60	-153	-66	-89	-154	144	111
61	-22	-22	-220	-242	147	35
62	110	-22	165	339	-203	-78
63	110	-66	361	-296	-91	-78