HARDWARE IMPLEMENTATION OF GOLD'S ALGORITHM FOR RENDEZVOUS IN ADAPTABLE FH COGNITIVE RADIO NETWORKS

Curtis Medve

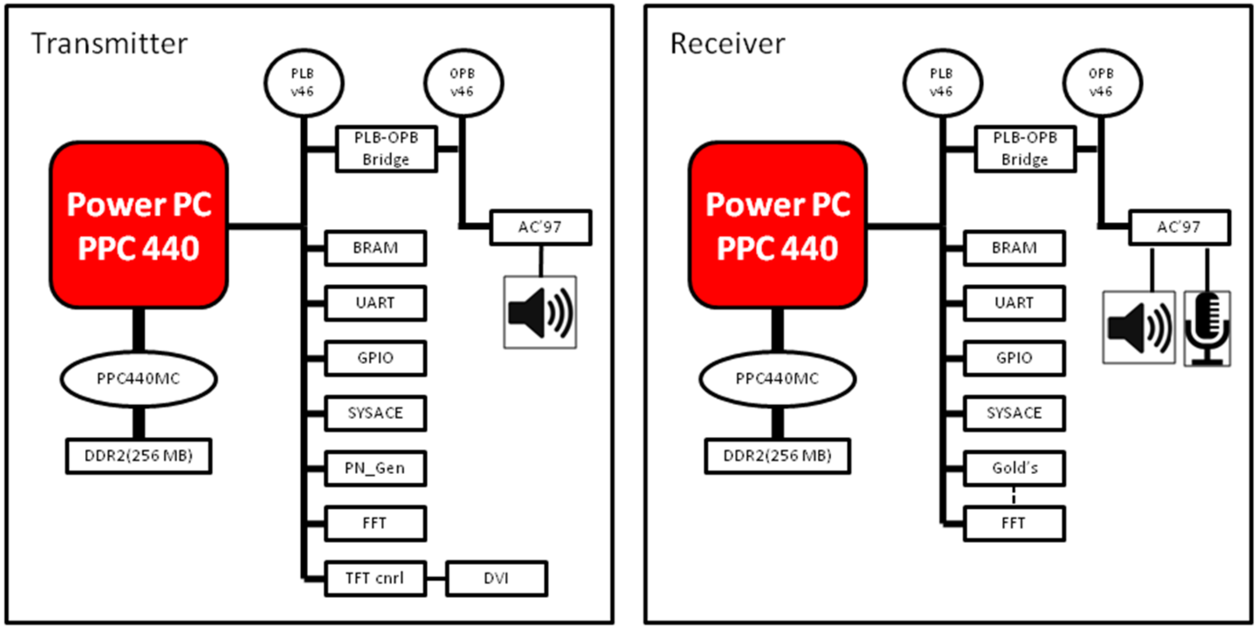
Mike Seery

Overview

The AFIT of Today is the Air Force of Tomorrow.

- Introduction
- Gold's Algorithm
- Methodology
- Hardware Design
- Results
- Future Work

Introduction

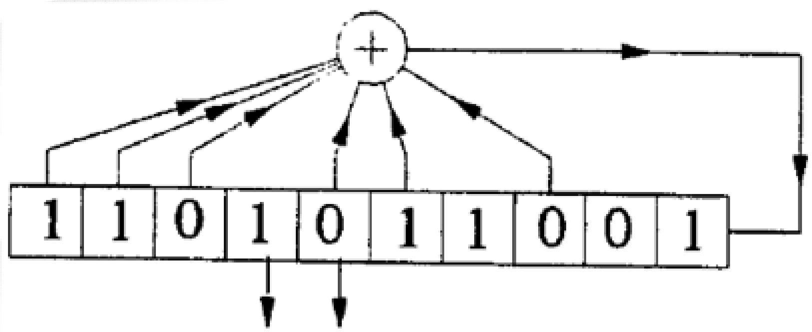


Gold's Algorithm (1)

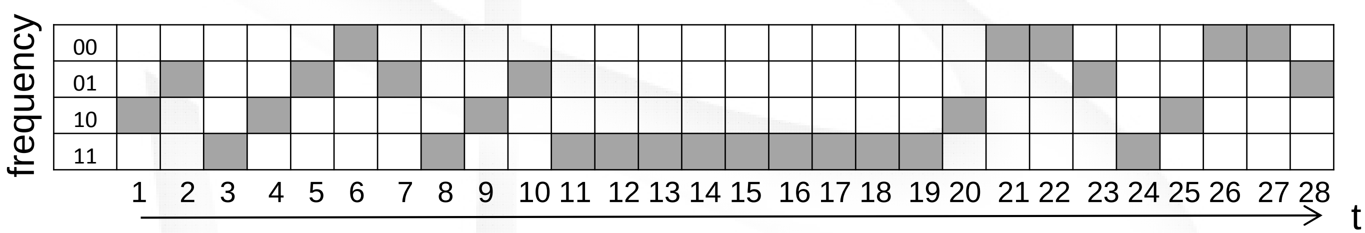
The AFIT of Today is the Air Force of Tomorrow.

Robert Gold Comm Systems Inc.
US Patent No.7,386,026 B1
Date of Patent: Jun.10, 2008.

- Linear Feedback Shift Register
- Matrix Representation

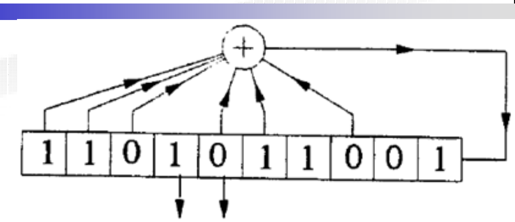


Target Configuration



11010110011011111111000110001

Gold’s Algorithm (2)



Target Configuration

- Foreknowledge
  - Feedback tap positions, $\mathbf{a} = (a_1 \quad a_2 \quad \dots \quad a_{10})$
  - Number of frequency taps, n

- Step1: Create transition Matrix

$$Ex: \mathbf{a} = (1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0)$$

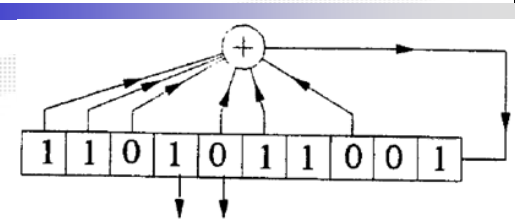
$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & a_4 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & a_5 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & a_6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & a_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & a_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & a_9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & a_{10} \end{pmatrix}$$

This matrix shows that if we have a given state of the LFSR at a given pulse, **z**, the next pulse state is given by:
[z1 z2 … z10] x A = [z2 z3 … z11]

This matrix operations allows us to calculate the state of any LFSR at ‘t’ pulses away from the original state by, **z** x A^t.

Gold's Algorithm (3)

The AFIT of Today is the Air Force of Tomorrow.



Target Configuration

- Step 2: Find time of arrivals of critical freq

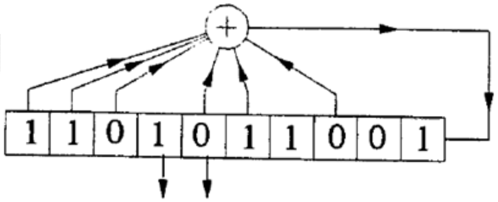
Seq: 1101011001101111111111000110001

Let the first occurrence be at t=0. This gives that the critical frequency occurs at t = 0, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20

- Step 3: Collect Matrix columns

We already stated, $\mathbf{z} \times A^t$, so now we let $\mathbf{z}_{Tn} = \mathbf{z}_0 \times A^{Tn}$ where Tn is a set of times of arrivals that step 2 provided and $\mathbf{z}_{Tn}$ is the state of the vector at Tn.

Gold's Algorithm (4)



Target Configuration

- Step 3 (Cont): Collect Matrix columns

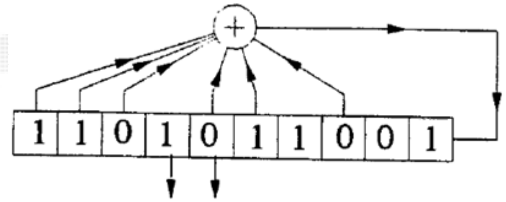
  Fun fact: We don't really care about the actual state or the original frequency tap points. We care about their relationship.

  As such, let us state that Bit1 is the first frequency tap. This will lead us to:

$$\begin{pmatrix} z_1 & z_2 & \dots & z_{10} \end{pmatrix} \times A^{T_n}(1) = z_1$$

Gold's Algorithm (5)

The AFIT of Today is the Air Force of Tomorrow.



Target Configuration

- Step 3 (Cont): Collect Matrix columns

  Collect all the first columns for each critical occurrence:

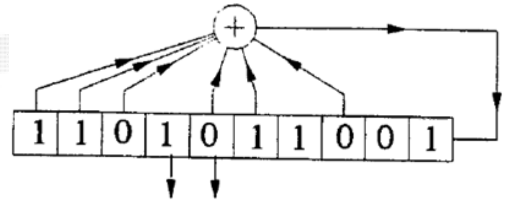
$$A^4(1) = \begin{pmatrix}0\\0\\0\\0\\1\\0\\0\\0\\0\\0\end{pmatrix} A^7(1) = \begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\1\\0\\0\end{pmatrix} A^8(1) = \begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\0\\1\\0\end{pmatrix} A^9(1) = \begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\\1\end{pmatrix} A^{10}(1) = \begin{pmatrix}1\\1\\1\\0\\1\\1\\0\\1\\0\\0\end{pmatrix}$$

$$A^{11}(1) = \begin{pmatrix}0\\1\\1\\1\\0\\1\\1\\0\\1\\0\end{pmatrix} A^{12}(1) = \begin{pmatrix}0\\0\\1\\1\\1\\0\\1\\1\\0\\1\end{pmatrix} A^{13}(1) = \begin{pmatrix}1\\1\\1\\1\\0\\0\\0\\0\\1\\0\end{pmatrix} A^{14}(1) = \begin{pmatrix}0\\1\\1\\1\\1\\0\\0\\0\\0\\1\end{pmatrix} A^{15}(1) = \begin{pmatrix}1\\1\\0\\1\\0\\0\\0\\1\\0\\0\end{pmatrix}$$

Which gives a system of equations:

$$\begin{pmatrix} z_1 & z_2 & \dots & z_{10} \end{pmatrix} \times \begin{pmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1\\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1\\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1\\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1\\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0
\end{pmatrix}$$

Gold's Algorithm (6)



Target Configuration

- Step 4 : Translate and Transpose

  What can we do with this?

$$(z_1 \quad z_2 \quad \dots \quad z_{10}) \times \begin{pmatrix} 0&0&0&0&1&0&0&1&0&1\\ 0&0&0&0&1&1&0&1&1&1\\ 0&0&0&0&1&1&1&1&1&0\\ 0&0&0&0&0&1&1&1&1&1\\ 1&0&0&0&1&0&1&0&1&0\\ 0&0&0&0&1&1&0&0&0&0\\ 0&0&0&0&0&1&1&0&0&0\\ 0&1&0&0&1&0&1&0&0&1\\ 0&0&1&0&0&1&0&1&0&0\\ 0&0&0&1&0&0&1&0&1&0 \end{pmatrix} = (z_1 \quad z_2 \quad \dots \quad z_{10}) \times \begin{pmatrix} 1&1&1&1&1&1&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0 \end{pmatrix}$$

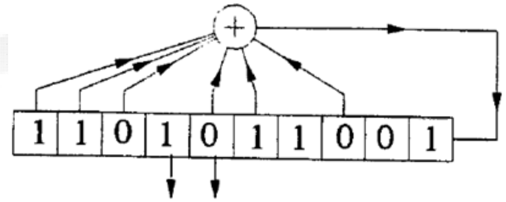
Subtracting away the right side we see that,

$$(z_1 \quad z_2 \quad \dots \quad z_{10}) \times \begin{pmatrix} 1&1&1&1&0&1&1&0&1&0\\ 0&0&0&0&1&1&0&1&1&1\\ 0&0&0&0&1&1&1&1&1&0\\ 0&0&0&0&0&1&1&1&1&1\\ 1&0&0&0&1&0&1&0&1&0\\ 0&0&0&0&1&1&0&0&0&0\\ 0&0&0&0&0&1&1&0&0&0\\ 0&1&0&0&1&0&1&0&0&1\\ 0&0&1&0&0&1&0&1&0&0\\ 0&0&0&1&0&0&1&0&1&0 \end{pmatrix} = 0 \quad \Rightarrow$$

Transposing this equation gives,

$$\begin{pmatrix} 1&0&0&0&1&0&0&0&0&0\\ 1&0&0&0&0&0&0&1&0&0\\ 1&0&0&0&0&0&0&0&1&0\\ 1&0&0&0&0&0&0&0&0&1\\ 0&1&1&0&1&1&0&1&0&0\\ 1&1&1&1&0&1&1&0&1&0\\ 1&0&1&1&1&0&1&1&0&1\\ 0&1&1&1&0&0&0&0&1&0\\ 1&1&1&1&1&0&0&0&0&1\\ 0&1&0&1&0&0&0&1&0&0 \end{pmatrix} \times \begin{pmatrix} z_1\\ z_2\\ z_3\\ z_4\\ z_5\\ z_6\\ z_7\\ z_8\\ z_9\\ z_{10} \end{pmatrix} = \begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0 \end{pmatrix}$$

Gold's Algorithm (7)



Target Configuration

- Step 5: Matrix Reduction

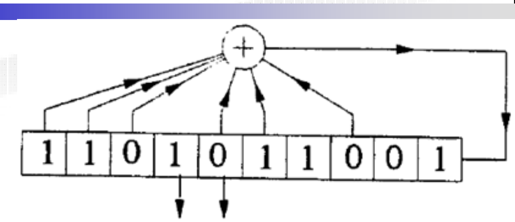
  Perform binary row reduction to obtain canonical form

$$\begin{pmatrix} 1&0&0&0&1&0&0&0&0&0\\ 1&0&0&0&0&0&0&1&0&0\\ 1&0&0&0&0&0&0&0&1&0\\ 1&0&0&0&0&0&0&0&0&1\\ 0&1&1&0&1&1&0&1&0&0\\ 1&1&1&1&0&1&1&0&1&0\\ 1&0&1&1&1&0&1&1&0&1\\ 0&1&1&1&0&0&0&0&1&0\\ 1&1&1&1&1&0&0&0&0&1\\ 0&1&0&1&0&0&0&1&0&0 \end{pmatrix} \times \begin{pmatrix} z_1\\ z_2\\ z_3\\ z_4\\ z_5\\ z_6\\ z_7\\ z_8\\ z_9\\ z_{10} \end{pmatrix} = \begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0 \end{pmatrix} \Rightarrow \begin{pmatrix} 1&0&0&0&0&0&0&0&0&1\\ 0&1&0&0&0&0&0&0&1&1\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&1&0\\ 0&0&0&0&1&0&0&0&0&1\\ 0&0&0&0&0&1&0&0&1&1\\ 0&0&0&0&0&0&1&0&0&1\\ 0&0&0&0&0&0&0&1&0&1\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0 \end{pmatrix} \times \begin{pmatrix} z_1\\ z_2\\ z_3\\ z_6\\ z_7\\ z_8\\ z_9\\ z_{10}\\ z_4\\ z_5 \end{pmatrix} = \begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0 \end{pmatrix}$$

Or

[z1; z2; z3; z4; z5; z6; z7; z8; z9; z10] = [z5; z4 + z5; 0; z4; z5; z4; z5; z4; z5; z5; z5]

Gold's Algorithm (8)



Target Configuration

- Step 6: Create Solution Space

  The possible solutions to the system of equations depend on z4 and z5. Specifically, it depend on the number of frequency taps needed.

[z1; z2; z3; z4; z5; z6; z7; z8; z9; z10] = [z5; z4 + z5; 0; z4; z5; z4; z5; z4; z5; z5; z5]

$z_4 z_5$	$z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10}$
0 0	0,0,0,0,0,0,0,0,0,0
0 1	1,1,0,0,1,1,0,1,1,1
1 0	0,1,0,1,0,1,1,0,0,0
1 1	1,0,0,1,1,0,1,1,1,1

Table 1: Vector Space of Candidate Solutions

Gold's Algorithm (9)

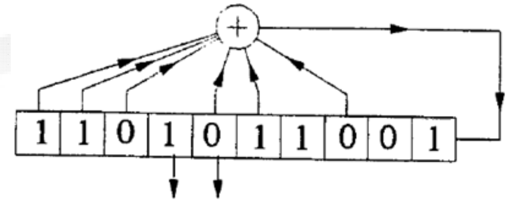
- Step 7: Determine Unique Solution

  Almost done! Of the 4 possible solutions, we can easily eliminate the all zero vector. With 3 possible solutions, we need to identify the shift relationships between them.

  Note that if $v_{0,1}$ is left shifted 1 digit it is equivalent to $v_{1,1}$. If any solution, can represent n other solutions, it is the Unique solution. $V_{0,1}$ contains both itself and $v_{1,1}$.

- Step 8: Frequency Taps

  The shift relationship between the Unique solution and the others solutions it contains give the tap points. Since $v_{1,1}$ was only 1 shift off, we get frequency taps at Bit 1 and Bit 2. Done!



Target Configuration

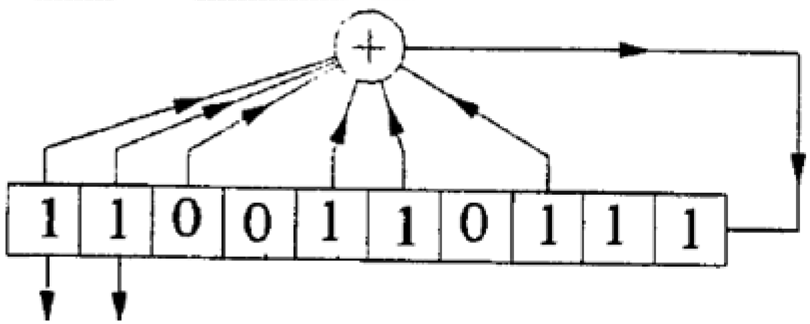
$z_4 z_5$	$z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10}$
0 0	0,0,0,0,0,0,0,0,0,0
0 1	1,1,0,0,1,1,0,1,1,1
1 0	0,1,0,1,0,1,1,0,0,0
1 1	1,0,0,1,1,0,1,1,1,1

Table 1: Vector Space of Candidate Solutions

Gold's Algorithm (10)

- Check

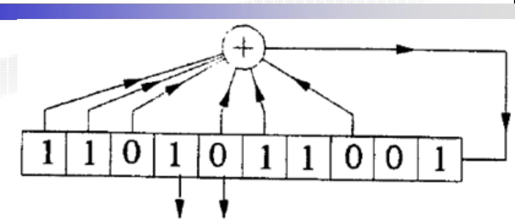
We have a unique solutions of [1100110111] and frequency taps at Bit 1 and Bit 2.



Alg. Seq: 110011011111111100011000111010111

Target Seq: 11010110011011111111111000110001

t=0



Target Configuration

$z_4 z_5$	$z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10}$
0 0	0,0,0,0,0,0,0,0,0,0
0 1	1,1,0,0,1,1,0,1,1,1
1 0	0,1,0,1,0,1,1,0,0,0
1 1	1,0,0,1,1,0,1,1,1,1

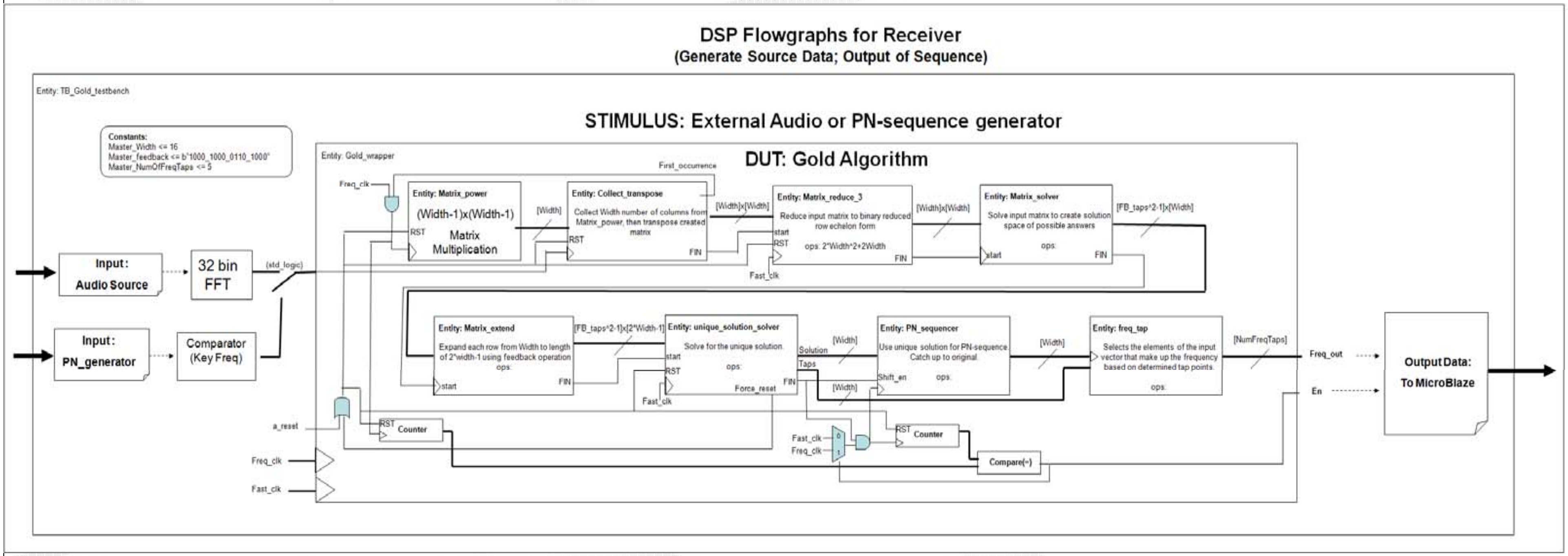
Table 1: Vector Space of Candidate Solutions

Methodology

FPGA based test bed for Gold’s Algorithm

- **Experimental Goal 1:** “Determine the physical constraints of hardware implementation of Gold’s Algorithm”
- **Experimental Goal 2:** “Compare the estimated time to rendezvous for hardware implementation versus Code-of-the-Day Method.”

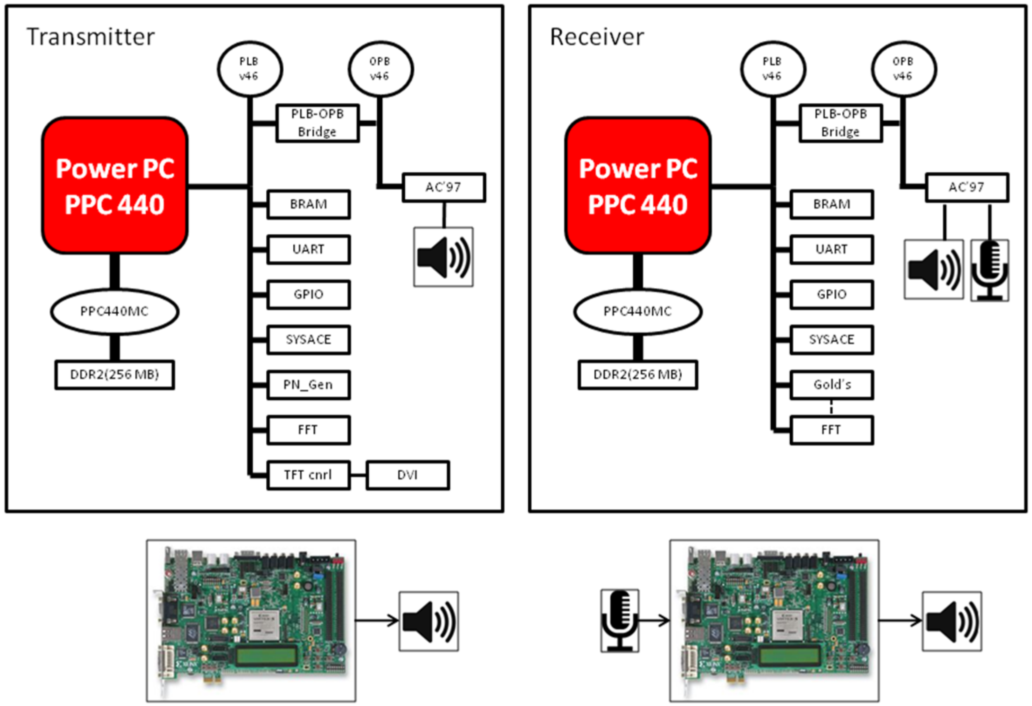
- Hardware Implementation



Hardware Design (2)

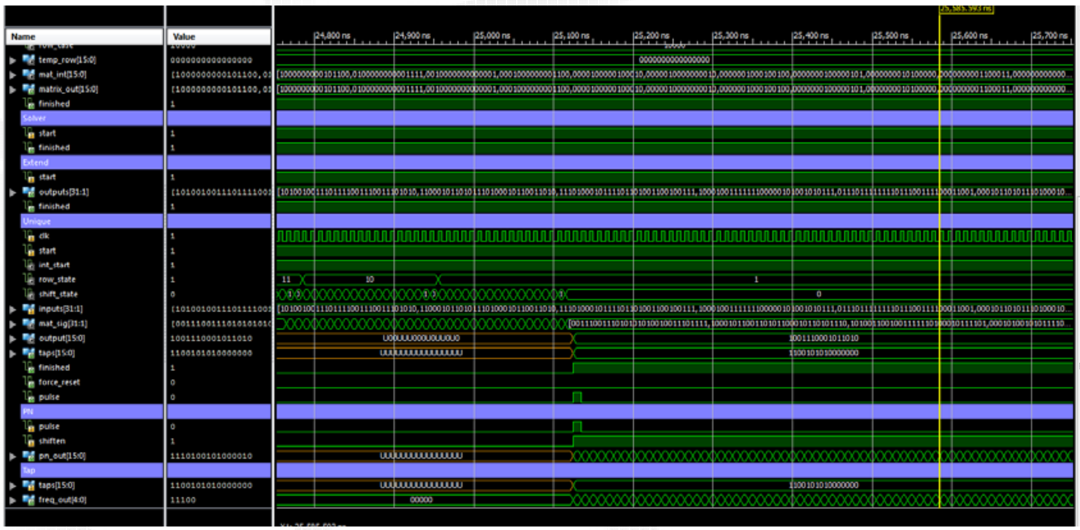
The AFIT of Today is the Air Force of Tomorrow.

- Design FPGA based test bed for Gold's Algorithm
  - Xilinx Virtex-5 – ML 507 development board (audio)
  - Xilinx ISE v13.2, application O.61xd, using XST



Results

- Virtex 5 audio test board



Results (2)

The AFIT of Today is the Air Force of Tomorrow.

- Physical Constraints

LFSR bit length	Number of Tap bits: 4	5	6	7	8	9	10
8	1.6688	1.8271	2.2062	2.3125	---	---	---
12	2.925	3.1229	3.1208	3.4125	3.8271	4.4979	5.7542
16	4.0875	4.2729	4.4396	4.6667	5.0333	5.8958	7.2854
20	5.6375	5.6521	5.9125	6.1563	6.5542	7.4542	8.7729
24	7.6083	7.7458	7.9625	8.1458	8.6083	9.6813	11.246
28	9.8187	9.8854	10.094	10.352	10.827	11.817	13.523
32	12.067	12.379	12.458	13.048	13.223	14.323	15.756
36	14.996	15.16	15.527	15.931	16.679	17.483	18.944

Virtex 5 Slice LUT Utilization (%)

LFSR bit length	Number of Tap bits: 4	5	6	7	8	9	10
8	1.0417	1.0854	1.1604	1.3062	---	---	---
12	1.8021	1.8458	1.9229	2.0667	2.3458	2.8896	3.9667
16	2.8396	2.8833	2.9604	3.1083	3.3917	3.9271	5.0042
20	4.1312	4.1771	4.2521	4.6042	4.675	5.2188	6.2958
24	5.6896	5.7354	5.8104	5.9542	6.2333	6.7771	7.8542
28	7.5146	7.5583	7.6354	7.7792	8.0583	8.6021	9.6792
32	9.6188	9.6625	9.7396	9.8833	10.163	10.706	11.783
36	11.977	12.021	12.098	12.242	12.521	13.065	14.142

Virtex 5 Slice Register Utilization (%)

Results (3)

The AFIT of Today is the Air Force of Tomorrow.

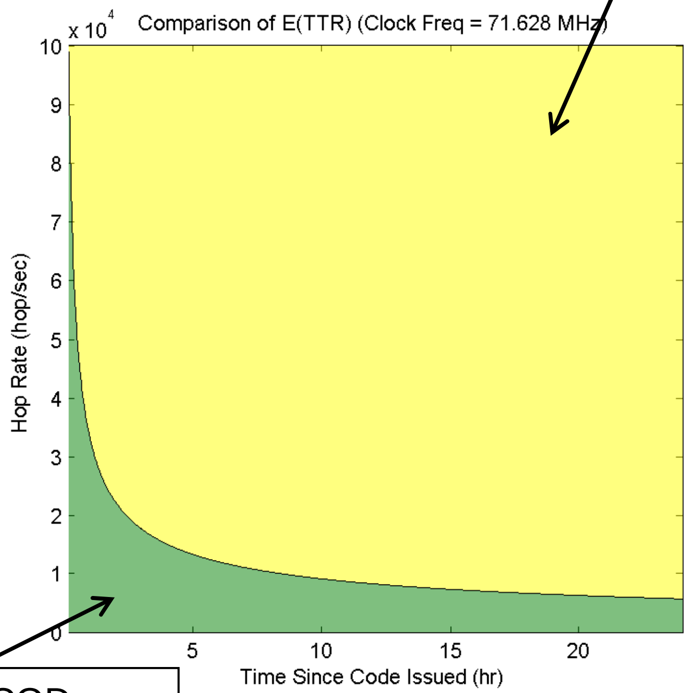
- Estimated time to rendezvous vs COD

Number of Tap bits / LFSR bit length

LFSR bit length	4	5	6	7	8	9	10
8	218.866	223.564	218.150	218.866	- - -	- - -	- - -
12	194.590	181.291	173.974	175.131	174.978	164.042	136.799
16	153.445	150.966	148.170	148.324	149.880	146.306	136.388
20	149.566	129.820	136.631	120.424	123.747	122.145	114.613
24	149.388	133.672	127.714	107.411	97.286	104.373	101.968
28	137.696	121.788	105.274	98.561	94.357	86.088	85.874
32	129.182	108.507	86.633	86.483	87.047	75.792	75.672
36	126.556	93.371	81.907	79.264	79.220	74.532	71.628

Maximum Clock Frequency (MHz)

Select Gold's algorithm



Select COD

Gold's algorithm provides faster rendezvous than Code-of-the-Day at high hop rates.

Future Work

- Discrete transmitter and receiver
- Migrate Gold's Algorithm to Rice University's Wireless Open Access Research Platform (Virtex 4) for RF implementation.
- Perform hardware-in-the-loop testing with Air Force Research Lab's Dynamic Spectrum Emulator (DYSE) to determine tolerance to interference.

Questions?

Curtis Medve
curtis.medve@afit.edu

Mike Seery
michael.seery@afit.edu