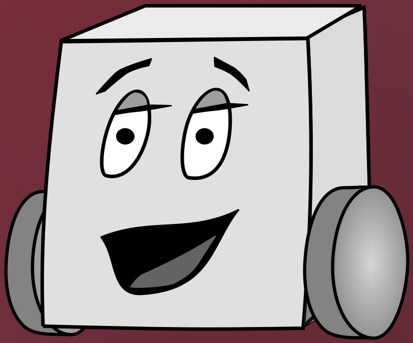


E11 Lecture 7: Gold Codes

Profs. David Money Harris & Sarah Harris

Fall 2011

Lab Notes

- Pick up your chassis
  - Several are still being printed
  - Printer garbled 4– let us know if yours is missing
- Please read your lab instructions before attending lab
- Remember to wear suitable machine shop attire this week
  - No open-toed shoes
  - No loose garmets
  - Long hair tied back

Outline

- Gold Code Overview
- Shift Register Sequences
- Gold Code Generation
- Gold Code Detection
- Applications

Overview

- Gold Codes are sequences of 0's and 1's
  - Commonly used in communications systems
    - Notably GPS and cell phones
  - Invented by Dr. Robert Gold in 1967
  - Easy to generate in hardware or software
  - Have characteristics resembling random noise
  - Minimally jam other Gold codes transmitted by other sources

Applications

- GPS
  - Multiple satellites transmit information simultaneusly at the same frequency
  - Receiver can pick out the signals from the individual satellites because each has a unique Gold code
- Your robot will seek beacons flashing different Gold codes
  - Identify the desired beacon by recognizing its code
  - Even if your phototransistor sees multiple interfering beacons
  - PS3: Gold Code Generation; PS6: Gold Code Detection

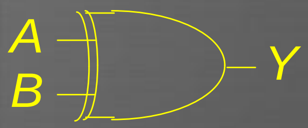
Mathematical Foundations

- Gold codes based on
  - XOR
  - Shift registers

XOR Review

- XOR of 2 inputs is TRUE if exactly one input is TRUE
- XOR of many inputs is TRUE if an ODD # of inputs are TRUE
- XOR is called a *linear* function

XOR

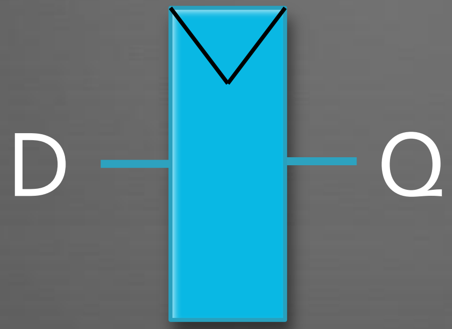


$Y = A \oplus B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

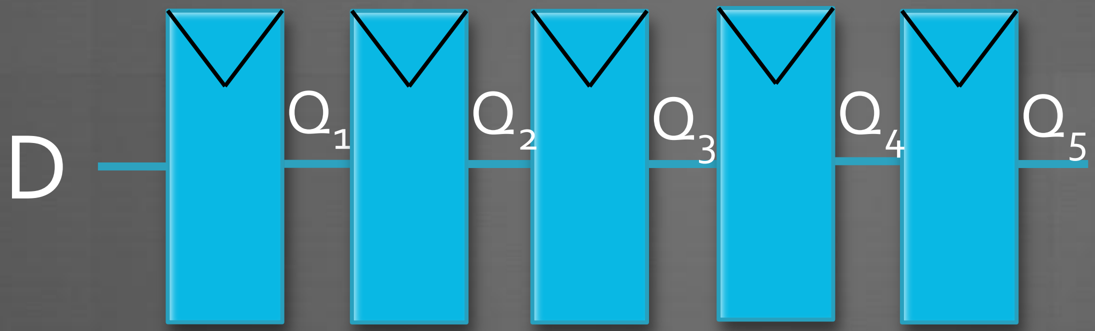
Register

- A register copies its input D to its output Q on each step



Shift Register

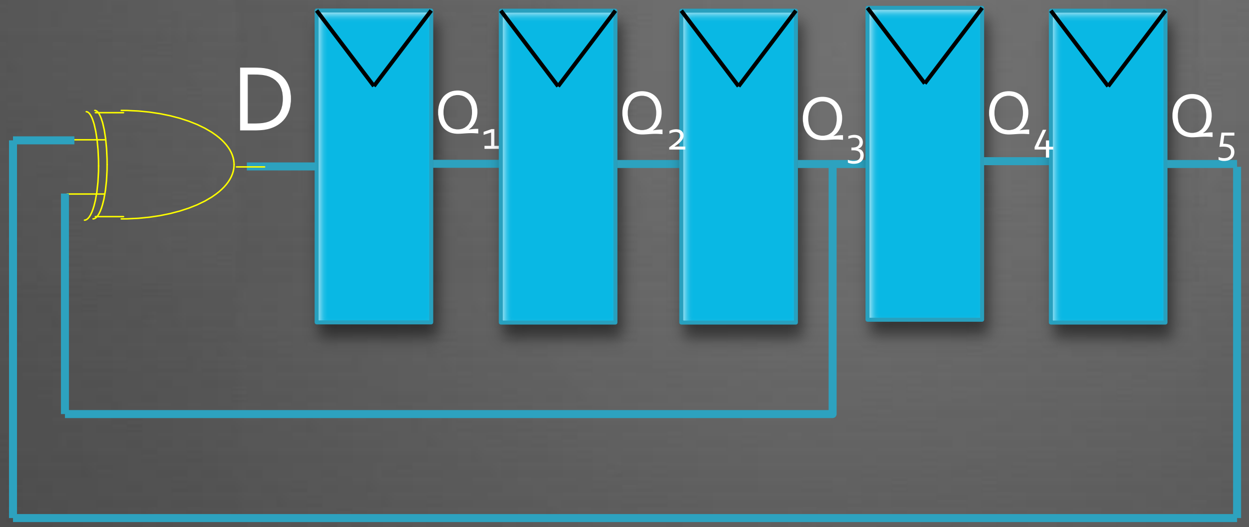
- A shift register shifts all of its bits right each step



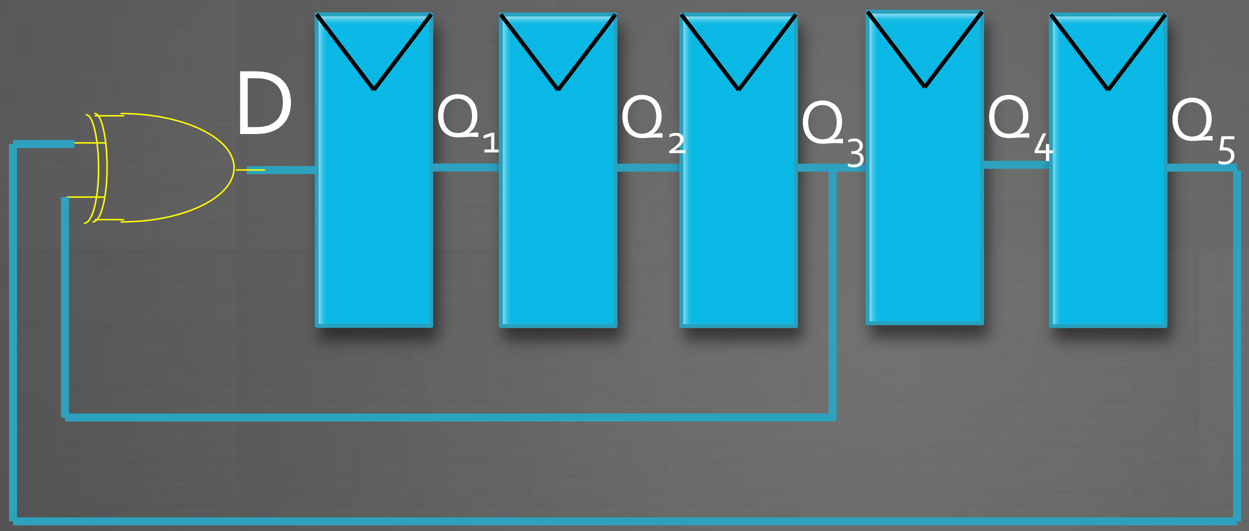
D	Q1	Q2	Q3	Q4	Q5
1	0	0	0	1	0
1	□	□	□	□	□
1	□	□	□	□	□

Linear Feedback Shift Register

- Linear Feedback Shift Register (LFSR)
  - Feeds XOR of certain bits back to input D



LFSR Operation



Step	Q1	Q2	Q3	Q4	Q5
0	0	0	0	0	1
1					
2					
3					
4					

Taps and Seeds

- Bits fed back are called the *taps*
  - LFSR taps are described by a *characteristic polynomial*
- Ex: $1 + x^3 + x^5$
  - Taps in columns 3 and 5
  - 1 is not a tap but corresponds to the input to the first bit x^0
- The initial contents of the LFSR are called the *seed*
  - Ex: 00001
  - If the seed is all 0’s, [redacted]

Complete Sequence

Step	Q1	Q2	Q3	Q4	Q5
0	0	0	0	0	1
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	1	0	0	1	0
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					

Step	Q1	Q2	Q3	Q4	Q5
16	0	0	1	1	1
17	0	0	0	1	1
18	1	0	0	0	1
19	1	1	0	0	0
20	0	1	1	0	0
21	1	0	1	1	0
22	1	1	0	1	1
23	1	1	1	0	1
24	0	1	1	1	0
25	1	0	1	1	1
26	0	1	0	1	1
27	1	0	1	0	1
28	0	1	0	1	0
29	0	0	1	0	1
30	0	0	0	1	0
repeat	0	0	0	0	1

Shift Register Sequence

- A *shift register sequence* is the pattern in the msb

Step	Q1	Q2	Q3	Q4	Q5
0	0	0	0	0	1
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	1	0	0	1	0
5	0	1	0	0	1
6	1	0	1	0	0
7	1	1	0	1	0
8	0	1	1	0	1
9	0	0	1	1	0
10	1	0	0	1	1
11	1	1	0	0	1
12	1	1	1	0	0
13	1	1	1	1	0
14	1	1	1	1	1
15	0	1	1	1	1
16	0	0	1	1	1
17	0	0	0	1	1
18	1	0	0	0	1
19	1	1	0	0	0
20	0	1	1	0	0
21	1	0	1	1	0
22	1	1	0	1	1
23	1	1	1	0	1
24	0	1	1	1	0
25	1	0	1	1	1
26	0	1	0	1	1
27	1	0	1	0	1
28	0	1	0	1	0
29	0	0	1	0	1
30	0	0	0	1	0

Sequence: 1000010010110011111000110111010

Maximal Length Sequences

- This is an example of a maximal length shift register seq.
  - Repeats after 31 = 2^5-1 steps
- In general, an *N*-bit MLSRS repeats after steps
- Not all characteristics polynomials produce MLSRSs

Runs of 0’s and 1s

- 10000100101100111110001101110 10
- □ run of length 5
- □ run of length 4
- □ runs of length 3
- □ runs of length 2
- □ runs of length 1
- All MLSRS have this distribution
  - Consistent with the statistics of random bit sequences

Seeding

- Different seeds give shifted version of the sequence

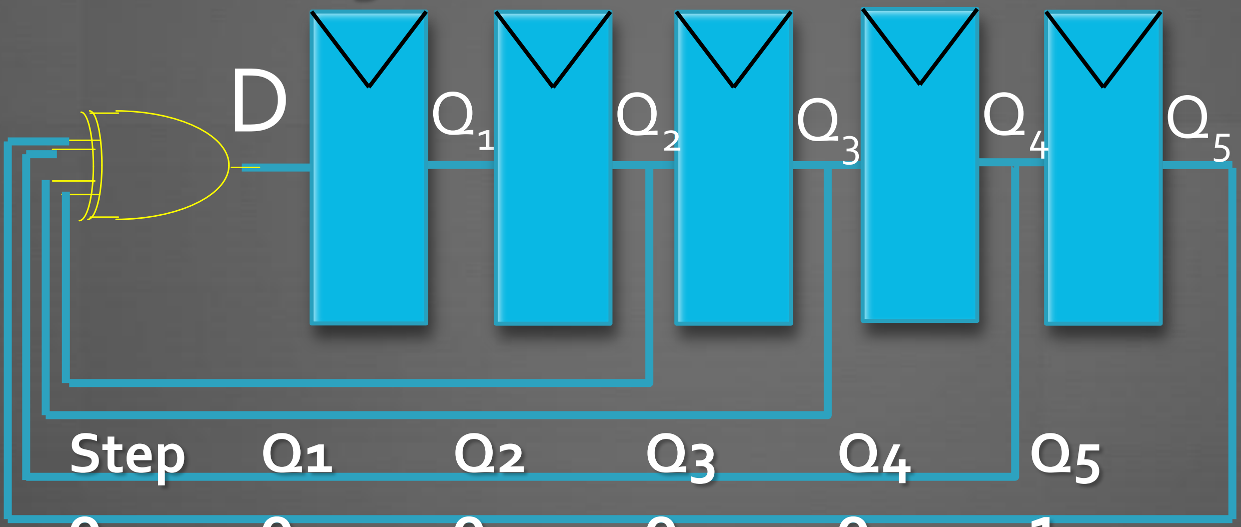
Step	Q1	Q2	Q3	Q4	Q5
0	0	0	0	0	1
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	1	0	0	1	0
5	0	1	0	0	1
6	1	0	1	0	0
7	1	1	0	1	0
8	0	1	1	0	1
9	0	0	1	1	0
10	1	0	0	1	1
11	1	1	0	0	1
12	1	1	1	0	0
13	1	1	1	1	0
14	1	1	1	1	1
15	0	1	1	1	1

Step	Q1	Q2	Q3	Q4	Q5
16	0	0	1	1	1
17	0	0	0	1	1
18	1	0	0	0	1
19	1	1	0	0	0
20	0	1	1	0	0
21	1	0	1	1	0
22	1	1	0	1	1
23	1	1	1	0	1
24	0	1	1	1	0
25	1	0	1	1	1
26	0	1	0	1	1
27	1	0	1	0	1
28	0	1	0	1	0
29	0	0	1	0	1
30	0	0	0	1	0 ← Seed

Seed 00010: Sequence 0100001001011001111100011011101

Another MLSRS

- $1+x^2+x^3+x^4+x^5$ generates a MLSRS: 1000010110101000111011111001001



Step	Q1	Q2	Q3	Q4	Q5
0	0	0	0	0	1
1					
2					
3					
4					

Gold Codes

- Communication systems need a set of bit sequences that:
  - Are easy to generate with hardware or software
  - Have a low cross-correlation with other sequences in the set
    - Easy to tell the sequences apart even when corrupted by noise
- Gold Codes are such a class of 2^N-1 sequences of length 2^N-1
  - Formed by XORing MLSRSs generated by different taps
  - Each seed gives a different Gold code
  - Each code is quite different than the others

Naming a Gold Code

- To uniquely define a Gold code:
  - State characteristic polynomial for the two LFSRs
  - State seed for the second LFSR
  - Always use a seed of 00…001 for the first LFSR
- Example: GC($1+x^2+x^3+x^4+x^5$, $1+x^3+x^5$, 00011)
- There are 2^N-1 Gold codes in a family
  - Defined by the different possible seeds (except 00…000)

5-bit Gold Code Examples

- GC($1+x^2+x^3+x^4+x^5$, $1+x^3+x^5$, 00001)

1000010110101000111011111001001 ($1+x^2+x^3+x^4+x^5$ seed 00001)

xor 1000010010110011111000110111010 ($1+x^3+x^5$ seed 00001)

0000000100011011000011001110011

- GC($1+x^2+x^3+x^4+x^5$, $1+x^3+x^5$, 00010)

1000010110101000111011111001001 ($1+x^2+x^3+x^4+x^5$ seed 00001)

xor 0100001001011001111100011011101 ($1+x^3+x^5$ seed 00010)

1100011111110001000111100010100

Gold Code Detection

- Read bit sequence
- Compare detected sequence with known Gold Codes
  - Use correlation: all possible dot products
  - Highest correlation indicates detected Gold Code

Dot Product

- *Dot product* of two binary sequences

  #of positions where bits match –

  # of positions where bits mismatch

- Ex: 110010 • 101010

  1 1 0 0 1 0

  1 0 1 0 1 0

  ■ ■ ■ ■ ■ ■

  -> dot product is ■

Dot Product Significance

- Dot product measures similarity of two sequences
  - Large positive dot product indicates strong similarity
  - Large negative dot product indicates nearly all bits differ
  - Dot product near 0 indicates two sequences are uncorrelated
  - Dot product of *l*-bit sequence with itself is *l*

Dot Products of SRS

- Example:

```
      1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0 (1 + x3 + x5 seed 00001)
dot   0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 (1 + x3 + x5 seed 00010)
      _____________________________________________________________
=     -1 -1 1 1 1 -1 -1 1 -1 -1 -1 1 -1 1 -1 1 1 1 1 -1 1 1 -1 1 -1 -1 1 1 -1 -1 -1
```

matches - mismatches

- Dot product is

Correlation

- *Cross-correlation* of two sequences
  - Measure of the similarity of the sequences when one is shifted by varying amounts.
  - Take the dot product of one sequence with each shifted version of the other
- *Autocorrelation*
  - Cross-correlation of a sequence with itself.

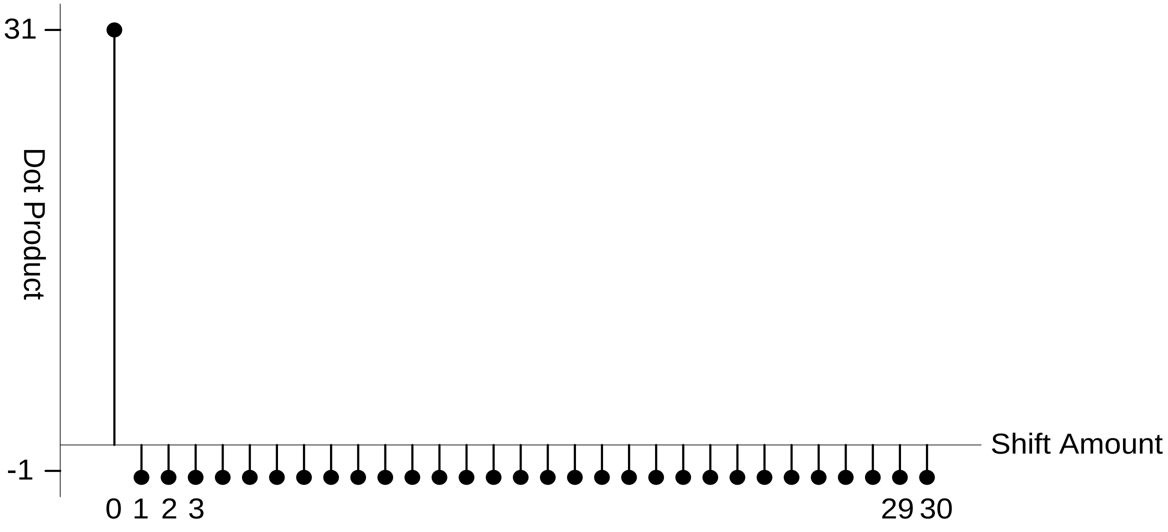
Autocorrelation Example

- 110010 • 110010 = 6 (shift by 0)
- 110010 • 011001 = -2 (shift by 1)
- 110010 • 101100 = -2 (shift by 2)
- 110010 • 010110 = 2 (shift by 3)
- 110010 • 001011 = -2 (shift by 4)
- 110010 • 100101 = -2 (shift by 5)

- Autocorrelation:6, -2, -2, 2, -2, -2

SRS Autocorrelation

- A MLSRS has an autocorrelation of 2^N-1 at an offset of 0
  - Autocorrelation of -1 at all other offsets



- Hence the MLSRS has characteristics of random noise

Pseudo-Random Bit Sequence

- MLSRS is also called a *pseudo-random bit sequence* (PRBS)
  - About half the bits are 0's and half 1's
  - Run length distribution consistent with randomness
  - Autocorrelation consistent with randomness
  - But sequence is deterministic and easy to generate with XOR

Gold Code Cross-Correlation

- A Gold Code has a correlation of 2^N-1 with itself
  - But a relatively low correlation with other codes in the family
  - Maximum cross-correlation is $2^{(N+1)/2} + 1$
- Thus, it is easy to detect the code by correlating
  - Even in the face of noise that flip some of the bits
- For our 5-bit code, correlation is 31 with itself
  - ≤ +7/-9 with other Gold codes
  - Called a *Hamming distance* of 31-9 = 22 between codes

Gold Code Correlation

- Correlation: Gold Code 1, Gold Code 2

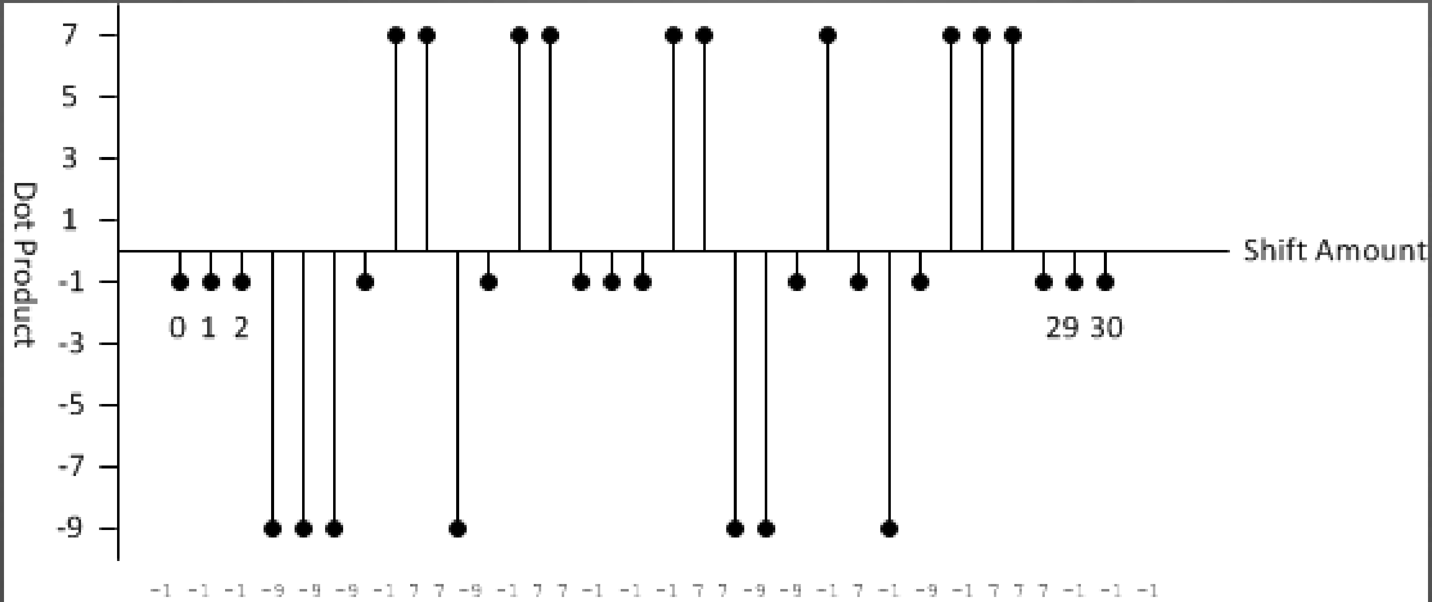
GC 1: 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 0 0 0 0 1 1 0 0 1 1 1 0 0 1 1
GC 2: 1 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 0 1 1 1 1 0 0 0 1 0 1 0 0

-1 -1 1 1 1 -1 -1 1 -1-1 -1 1 -1 1 -1 1 1 1 1 -1 1 1 -1 1 -1 -1 1 1 -1 -1 -1

shift =0, dot product = -1

Cross-Correlation

- Cross-correlation of
  - GC($1+x^2+x^3+x^4+x^5$, $1+x^3+x^5$, 00001)
  - GC($1+x^2+x^3+x^4+x^5$, $1+x^3+x^5$, 00010)



Application: Beacons

- Eight LED beacons on the E11 playing field
  - Beacon *b* (*b* = 1…8) flashes GC($1+x^2+x^3+x^4+x^5$, $1+x^3+x^5$, *b*)
  - 4 KHz data rate (250 microseconds / bit)
  - Sequence is inverted depending on team (white vs. green)
- Detect beacons using a phototransistor on your bot
  - Produces a voltage related to the light intensity
  - Principles of operation to be described later

Identifying a Beacon

1. Read 31 phototransistor samples at 4 KHz
2. Compute average value
3. Convert readings to binary by comparing to average
4. Correlate against each of 31 offsets for each of 8 beacons
5. If correlation exceeds a threshold, report beacon found
6. Improve accuracy by taking more than 31 samples

Application: GPS

- 24 satellites orbit earth
  - At least 6 are visible in the unobstructed sky at any time
- All satellites broadcast 10-bit Gold Codes
  - All share a 1.575 GHz carrier
  - 1.023 MHz code rate
    - 1023 bits / sequence -> repeats every 1 ms
  - Each satellite jams all of the others
  - Thermal noise exceeds strength of all satellites combined
  - But satellites are identified by correlation (!)
- 50 Hz data rate
  - Transmitted signal may be inverted based on data value

wikipedia.com

Application: CDMA

- Code Division Multiple Access (cell phones)
  - All phones transmit on all frequencies simultaneously
  - Each uses its own 15-bit (length 32767) Gold Code
  - Identify the phone by correlating against its Gold Code
- Developed by Qualcomm
  - Replaces Time Division Multiple Access
    - Where each user gets a time slot (TDMA)
  - Better quality reception when spectrum is not completely full
  - Central to 3G and 4G wireless systems