HARDWARE ACCELERATOR REPORT

TEAM MEMBERS:
Atluri Sai Prabhakar (saipraba@usc.edu)
Shashank (sshashan@usc.edu),
Shishir Madhusudan (madhusus@usc.edu)

Status: We have designed and synthesized the hardware accelerator for accelerating different CRC standards for the 4 cores of the Gold Chip Processor.

Application: In today's computer systems, various communication modules (Ethernet, WLAN, PCIe, USB, etc.) employing different CRC standards are integrated with Central Processor Unit. Hence, usage of dedicated hardware accelerator for each application specific CRC standard is beneficial.

Design Principle: We have used LFSR based CRC computation. The polynomials used corresponding to each type of CRC is as follows:

CRC TYPE	POLYNOMIAL
CRC5 for USB2.0 [crc5.v ; tb_crc5.v]	1+x^2+x^5
CRC16 for USB2.0 [crc16.v ; tb_crc16.v]	1+x^2+x^15+x^16
CRC for PCIe [crc32PCIe.v ; tb_crc32PCIe.v]	1+x^1+x^2+x^4+x^5+x^7+x^8+x^10+x^11+x^12+x^16+x^22+x^23+x^26+x^32
CRC for Ethernet [802.3 crc32.v ; tb_crc32.v]	1+x^1+x^2+x^4+x^5+x^7+x^8+x^10+x^11+x^12+x^16+x^22+x^23+x^26+x^32

DATA WIDTH is 64 in all cases.

Background Theory:

A. LFSR [Linear Feedback Shift Register]

These are n-bit counters exhibiting pseudo-random behavior, built from simple shift-registers with a small number of xor gates.

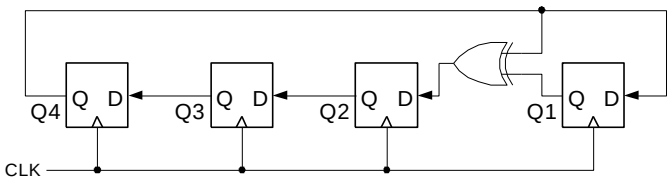
Used for:

- Random number generation
- Counters
- Error checking and correction

Advantages:

- Very little hardware
- High speed operation

Example 4-bit LFSR:



B. CRC (Cyclic Redundancy Check)

We have a msg polynomial M(x) of degree m and a generator poly G(x) of degree m.

- Let r(x) = remainder of $M(x)\ x^n$ / G(x)
- $M(x)\ x^n = G(x)p(x) + r(x)$; r(x) is of degree n

We got to investigate $(M(x)\ x^n - r(x))$ / G(x)

We send M(x) x^n – r(x)

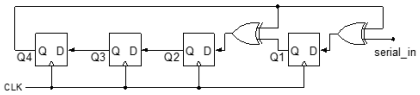
- m+n degree polynomial
- Then we divide by G(x) to check
- M(x) is just the m most significant coefficients, r(x) the lower m

n-bit Message is viewed as coefficients of n-degree polynomial over binary numbers

C. LINK BETWEEN LFSR AND CRC

The following figures illustration the process of generating CRC encoding using an LFSR.

CRC ENCODING:



1 0 1 1 0 0 1 0 0 0 0

0	0	0	0	
0	0	0	1	0 1 1 0 0 1 0 0 0 0
0	0	1	0	1 1 0 0 1 0 0 0 0
0	1	0	1	1 0 0 1 0 0 0 0
1	0	1	1	0 0 1 0 0 0 0
0	1	0	1	0 1 0 0 0 0
1	0	1	0	1 0 0 0 0
0	1	1	0	0 0 0 0
1	1	0	0	0 0 0
1	0	1	1	0 0
0	1	0	1	0
1	0	1	0	

Message sent:

1 0 1 1 0 0 1 1 0 1 0