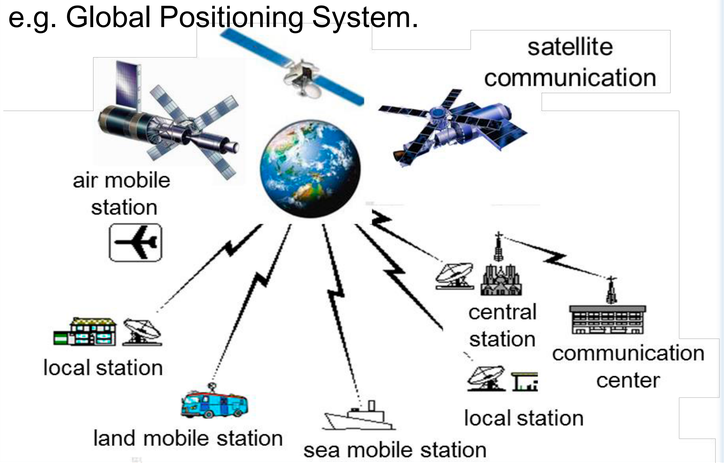
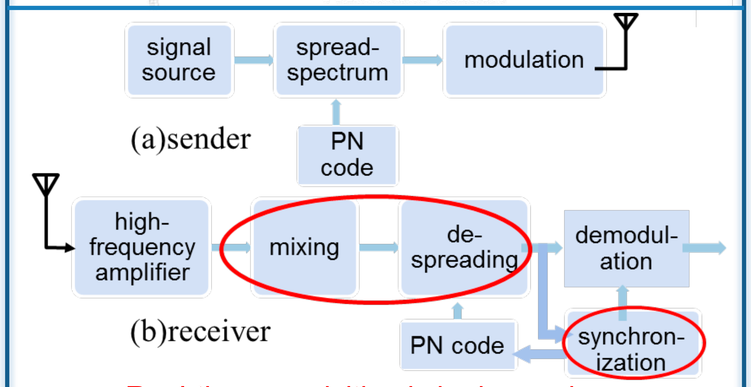
CUDA-Accelerated Acquisition of Spread Spectrum Signal in Satellite Communication

Ying Liu, Hongyuan Cui, Renliang Zhao, Yang Gao
School of Computer and Control, University of Chinese Academy of Sciences

Spread Spectrum Communication

Spread spectrum techniques deliberately spread a signal with a particular bandwidth into a signal with wider bandwidth. Due to more secure communication and enhanced resistance to natural interference, noise and jamming, spread spectrum signal has been used in many applications in satellite communication, e.g. Global Positioning System.





Real-time acquisition is in demand.

Acquisition Algorithm

```
for(int ch = 0; ch < # of channels; ch++){
    for(int i = 0; i < needLen; i++){
        PNcode[i] = pnData[(int)floor(fmod(((float)pnRate*
                    (1.0+(channels[ch])/ft*1)*i/fs),1023.0))];
        mix_I[i] = sin(2*pi*(f0+channels[ch])*i/fs+phase0)*sig[i];
        mix_Q[i] = cos(2*pi*(f0+channels[ch])*i/fs+phase0)*sig[i];
    }
    for(int j = 0; j < pointCodeN; j++){
        for(int t = 0; t < NL * NA; t++){
         mi_I[t] = localGold[t] * mix_I[pointCode*j + t];
         mi_Q[t] = localGold[t] * mix_Q[pointCode*j + t];
        }
        for(int k2 = 0; k2 < NL; k2++){
            for(int k1 = 0; k1 < N; k1++){
                tmpI = 0.0; tmpQ = 0.0; n = k1*sumL + k2*NA;
                for(int p = n; p < n + sumL; p++){
                    tmpI += mi_I[p]; tmpQ += mi_Q[p];
                }
                dataI[NL*N*j+k1+k2*N] = tmpI;
                dataQ[NL*N*j+k1+k2*N] = tmpQ;
            }
        }
    }
    ippsFFTFwd_CToC_32f(dataI, dataQ, yR, yI, spec, NULL);
    for(int e = 0; e < Nfft*NL*pointCodeN; e++){
        yR[e] = sqrt(yR[e] * yR[e] + yI[e] * yI[e]);
    }
    for( int q = 0; q<pointCodeN;q++){
        for(int t = 0; t < Nfft; t ++){
            for(int j = 0; j < NL; j ++){
                Rout[q*Nfft+t] += yR[q*Nfft*NL + j*Nfft + t];
            }
        }
    }
    // select the max value from Rout, then its position is acquired.
}
```

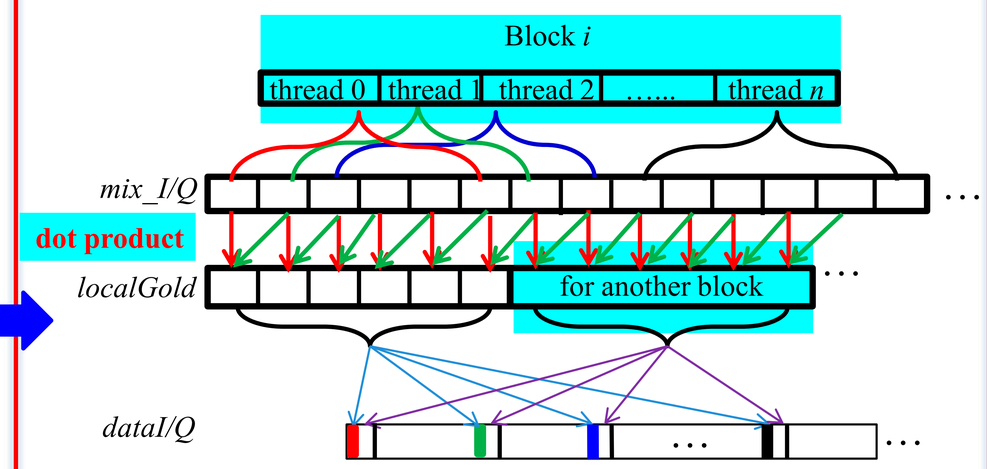
Computation core!

CUDA enabled Parallel Acquisition Algorithm

Compelxity: O(NL*NA*pointCodeN*channels)

Phase 1: Mixing (parallelism: *needLen*)

Each thread calculates three elements in three vectors: PNcode, branch I of the mixed signal, branch Q of mixed signal.



Phase 2: sliding correlation (parallelism: *pointCodeN*N*NL*)

- Each thread calculates a dot product of *sumL* pairs of points;
- Different threads in a block take different segments of mix_I/Q;
- Load the segment of mix_I/Q into the shared memory on an SM;
- All the threads in a block share a common segment of the PNcode, load the common part into the shared memory on an SM;

Phase 3: FFT-based acquisition (parallelism: *pointCodeN*Nfft*)

- Perform FFT on datal and dataQ by invoking CUFFT in batch mode;
- Each thread computes the modulus of NL signal points, and accumulates the moduli.

Performance Evaluation

- Six sets of data with 56M/s sampling rate from a real telemetry system
- 300 microseconds, each set containing 8192*2048 floats

#	# of channels	pointCodeN	NA	NL	N	sumL	pointCode
1	8	2074	6222	40	64	87	2
2	8	2810	5600	40	64	97	3
3	9	2334	7000	50	64	109	3
4	10	2667	8000	40	64	125	3
5	15	2280	11200	30	64	175	5
6	12	2340	9333	40	64	145	4

- **Platform description**
  - Intel ® Xeon® CPU, 2.40GHz
  - NVIDIA's Tesla K20 card, CUDA 4.0
- **Speedups on K20**

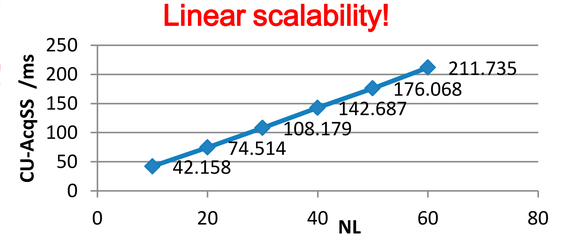
Real-time acquisition was achieved in most cases !

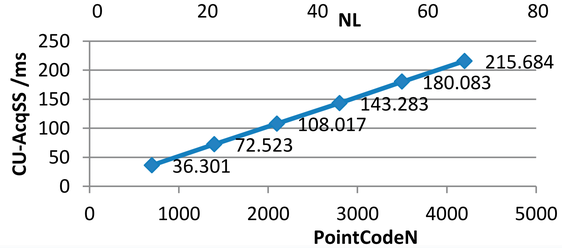
Data Set #	CPU (ms)	CU-AcqSS (ms)	Speedup
1	15,902	76.255	208.537
2	19,552	142.687	137.027
3	27,431	128.921	212.774
4	30,206	171.481	176.148
5	39,755	245.084	162.210
6	36,763	396.639	92.686

Counterpart: CPU version using Intel®IPP

- **Scalability when increasing *NL* and *PointCodeN***

Linear scalability!





Conclusion

- Identified the computational core of the acquisition algorithm, sliding correlation
- Proposed an efficient scheme to accelerate sliding correlation on CUDA
- Implemented the acquisition algorithm on CUDA
- Evaluated on 6 sets of data
- Up to 212X speedups were observed
- Good scalability was observed when varying the computation
- Real-time acquisition was achieved in most cases