THE UNIVERSITY of TENNESSEE KNOXVILLE

Reconfigurable Computing For Cholesky Decomposition

Depeng Yang, Gregory D. Peterson, and Husheng Li

Electrical Engineering and Computer Science

THE UNIVERSITY of TENNESSEE KNOXVILLE

Outline

- Background and motivation
- Cholesky decomposition procedure
- Architecture and performance
- Customized precision and error analysis
- Conclusions

Background

- Cholesky decomposition is a computationally expensive step in solving least square problems in signal processing.
- In compressed sensing it desires to speed up signal reconstruction algorithms, in which Cholesky decomposition is the key step.
- FPGAs provide an approach to speeding up computations.

Cholesky Decomposition

- Standard Cholesky decomposition is associated with square root and division operation. The heavy data dependency makes it very hard to obtain a speedup.

$$\mathbf{L}\,\mathbf{L}^{\mathbf{T}} = \mathbf{A}$$

```
for i = 1 to N do
  begin
    l(i, i)= SQRT(a(i, i));
    for j = i+1 to N do
      begin
        l(j, i)= a(j, i)/l(i, i);
       for k = i+1 to j do a(j, k) = a(j, k) - a(j,i)*a(k, i);
    end
  end
```

THE UNIVERSITY of TENNESSEE UT KNOXVILLE

Solving Linear Equations

- How to solve *Ax=b*, if $A^T=A$?

Solution: $Ax=b$ => $LL^T=b$ => $Lr=b$ => …)

Heavy Data Dependency !!!

- How to solve $L^Tx=r$ on FPGAs? Can we achieve a speedup using FPGAs?

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}$$

$$x_4 = r_4 / u_{44}$$
$$x_3 = (r_3 - u_{34}x_4) / u_{33}$$
$$x_2 = (r_2 - u_{24}x_4 - u_{23}x_3) / u_{22}$$
$$x_1 = (r_1 - u_{14}x_4 - u_{13}x_3 - u_{12}x_2) / u_{11}$$

Speedup Forward/Backward Substitutions

- *LDL* Cholesky decomposition:

  $A=LDL^T$ (*D: diagonal matrix*)

  -> $Ax=b$ => $LDL^Tx=b$ => $LDr=b$ => $L^Tx=r$

- By separating the divider, a speedup is achieved for forward/backward substitutions.

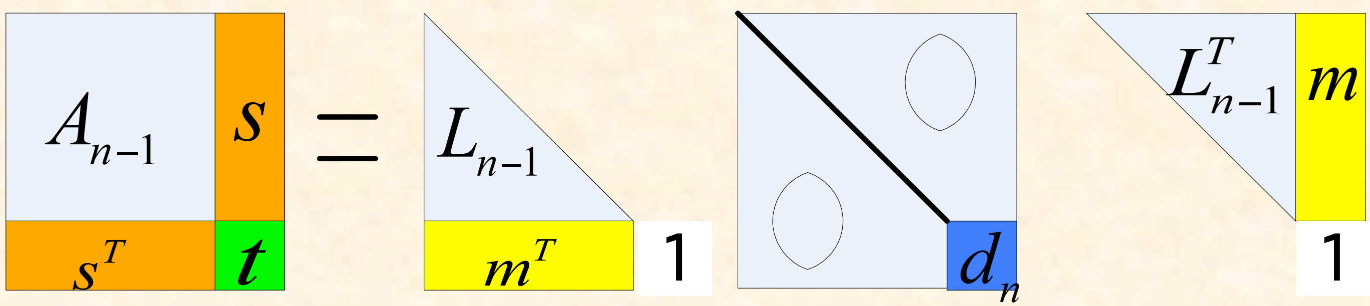
$$\begin{pmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}$$

$$\begin{aligned} x_4 &= r_4 \\ x_3 &= (r_3 - u_{34}x_4) \\ x_2 &= (r_2 - u_{24}x_4 - u_{23}x_3) \\ x_1 &= (r_1 - u_{14}x_4 - u_{13}x_3 - u_{12}x_2) \end{aligned}$$

Novel Cholesky Decomposition

- The original matrix A_n is partitioned into a 2x2 block matrix which consists of the matrix A_{n-1} , a column vector *s* and a scalar number *t*, whose Cholesky decomposition is given by:

$$A_n = \begin{pmatrix} A_{n-1} & s \\ s^T & t \end{pmatrix} = \begin{pmatrix} L_{n-1} & 0 \\ m^T & 1 \end{pmatrix} \begin{pmatrix} D_{n-1} & 0 \\ 0 & d_n \end{pmatrix} \begin{pmatrix} L_{n-1}^T & m \\ 0 & 1 \end{pmatrix}$$



Novel Cholesky Decomposition

- Assuming the decomposition of matrix A_{n-1} is known, giving:

$$A_{n-1} = L_{n-1} D_{n-1} L_{n-1}^T$$

- Then for factorizing A_n we just need to update the lower triangular matrix by solving lower triangular equations to obtain vector *m* and scale *g:*

$$L_{n-1} D_{n-1} m = s$$

$$d_n = t - s^T D_{n-1} s = t - \sum_{i}^{n-1} s_i^2 d_i$$

- **In sequence, starting with A_1 (the most up left element in A_n) the matrix A_n decomposition is calculated by iteratively solving triangular linear equations.**

Computation Sequence for Pipeline

- A pipeline is designed for solving lower triangular equations *LDs=b; (=>Lz=b; Ds=z;)* and computation sequence is illustrated in the table:

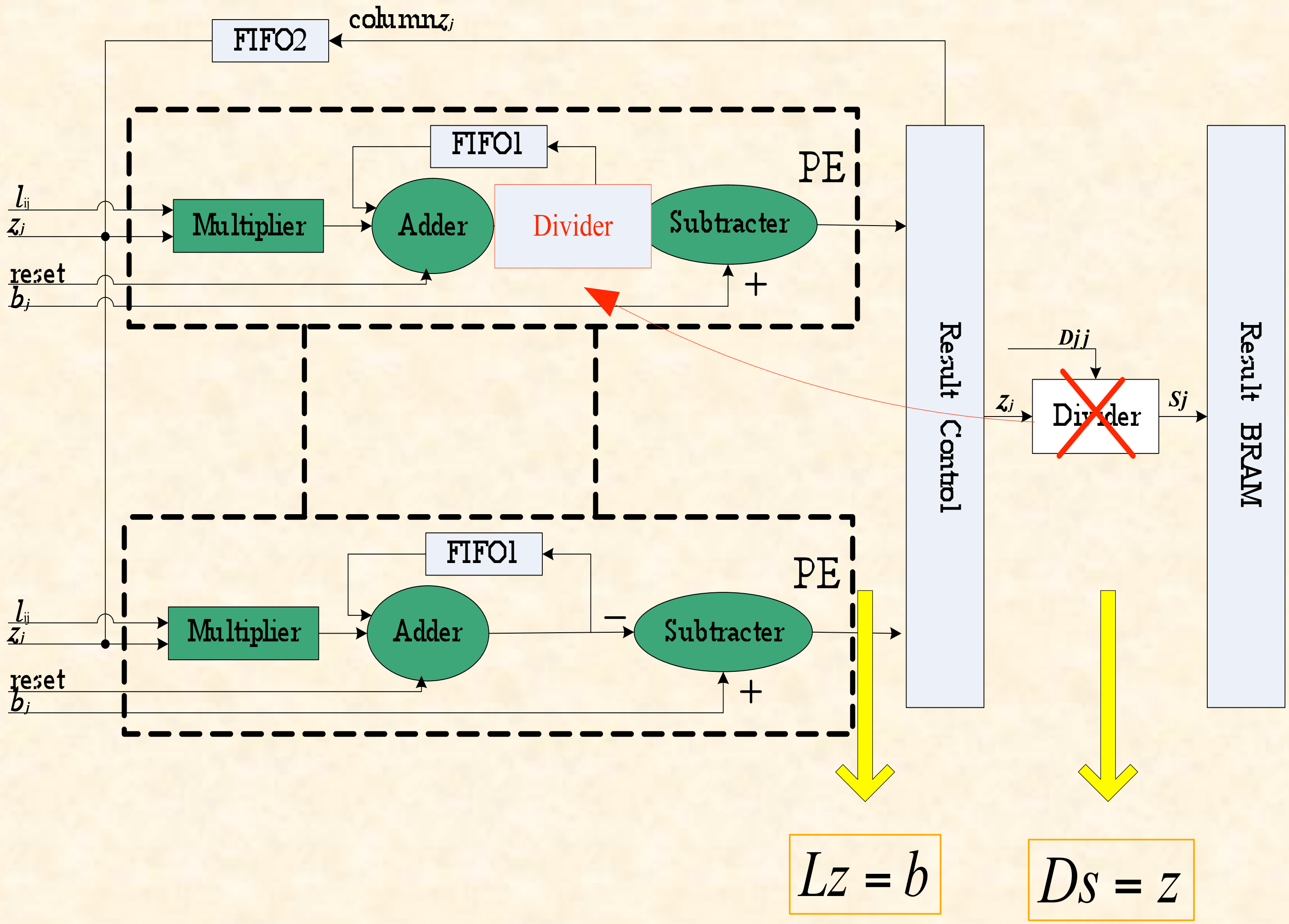
Step 1	Step 2	…	Step n-1
$(z_1=b_1)$ $z_2=b_1-l_{21}z_1$ $z_3=l_{31}z_1$ $z_4=l_{41}z_1$ …… $z_n=l_{n1}z_1$	$z_3=b_3-l_{31}z_2-z_3$ $z_4=-l_{41}z_2-z_4$ …… $z_n=-l_{n1}z_2-z_n$	. . .	$z_n=b_n-l_{n-1n-1}z_{n-1}-z_n$
$s_1=z_1/d_1$	$s_2=z_2/d_2$	.	$s_n=z_n/d_n$
Feed back z_2	Feed back z_3	.	

$$Lz = b$$

$$Ds = z$$

THE UNIVERSITY of TENNESSEE KNOXVILLE

Architecture of PEs



Performance

- Testing matrix size is 256x256 and we use 16 PEs.
- Xilinx XC5VSX95T-2 FPGA (containing 14720 slices and 640 DSP48 modules).
- C code running on the CPU with a Quad core 3GHz Intel Xeon X5450, 6144KB cache and 2GB memory.

Design	s20e8	s23e8 (single)	s32e11	s46e11	s52e11 (double)
Freq	*265MHz*	*255*	*220*	*206*	175
Slices	*19%*	*24%*	*34%*	*62%*	*73%*
DSP48	*7%*	*22%*	*24%*	*35%*	*47%*
CPU	140.4875*us*			145.9826*us*	
Interface	200MHz			100MHz	
FPGA	10.96 *us*			21.76 *us*	
speedup	~13			~7	

THE UNIVERSITY of TENNESSEE KNOXVILLE

Customized Precision

- The precision of the PEs is customized by adjusting mantissa bits with fixed exponential bits.
- Taking double precision (s52e11: 52bits mantissa and 11bits exponential) as a reference, the error of the result L and D matrices is defined as:

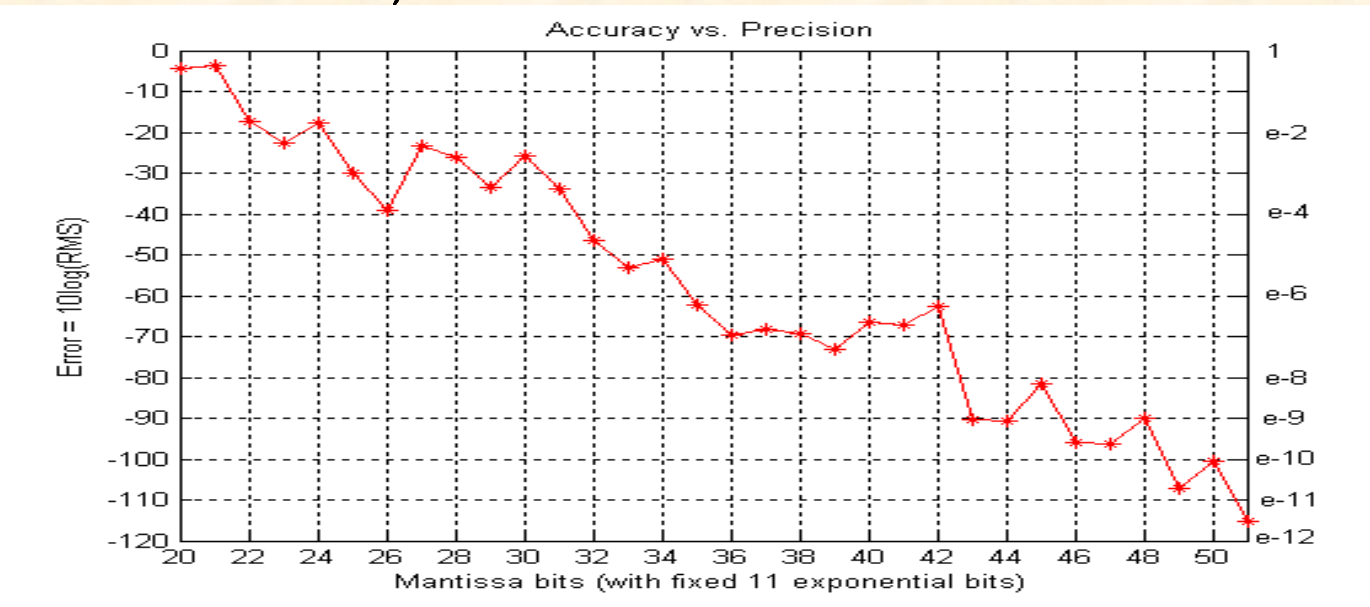
$$Error = 10\log\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}(\tilde{L}_{ij} - L_{ij})^2 / N + \sum_{i=1}^{N}(\tilde{D}_i - D_i)^2 / N}$$

- 1000 matrices for Cholesky decomposition with randomly distributed elements are tested and results are averaged.

THE UNIVERSITY of TENNESSEE UT KNOXVILLE

Customized Precision and Error

- The error is exponentially decreased while increasing mantissa bits, even though error is affected by the condition number of the original matrix.
- Tradeoffs: Lower precision leads to fewer hardware resource and potentially higher performance, but results into lower accurate matrices; and vice versa.



THE UNIVERSITY of TENNESSEE UT KNOXVILLE

Conclusions

- We adopt LDL decomposition to avoid division operations and corresponding long latency.
- We propose to use a novel Cholesky decomposition procedure in which by designing a single hardware triangular linear equation solver, the Cholesky decomposition is realized.
- Different precisions vs. error is analyzed.
- A good speedup has achieved on FPGAs compared with CPUs.
- Future work is to compare GPGPU performance with FPGAs.

THE UNIVERSITY of TENNESSEE
KNOXVILLE

Thanks !