Floating point units in hybrid FPGA using clock gating

BHUVANESH.N
M.E Vlsi design(PG scholar)
Srinivasan Engineering College
Perambalur-621212
nbhuvi@gmail.com

Abstract – Low power and high speed are the two major challenges in the VLSI circuit design. Subgraph extraction is a method to optimize coarse grained floating point units (FPUs) in a hybrid field-programmable gate array (FPGA), where the FPU consists of a number of interconnected floating point adders/ subtracters (FAs), multipliers (FMs), and wordblocks(WBs). The wordblocks consist of registers and lookup tables (LUTs) which can implement fixed point operations efficiently. Common subgraph extraction is used to determine the best mix of blocks within an FPU for increasing the speed of the system.By using clockgating method the power can be reduced. Finally, an optimized coarse-grained FPU by considering both architectural and system-level issues can be achieved and also the power of the floating point units is compared.

KeyTerms: Common subgraph extraction, field-programmable gate array (FPGA), floating point.

I.INTRODUCTION

In the hybrid FPGAs,coarse-grained elements such as memories and digitalsignal processors (DSPs) are embedded within a fine-grainedprogrammable fabric. Since coarse grained elements are less flexible, it is efficient for implementing specific word level operations. Improved speed and density can be achieved by incorporating embedded floating point units in an application which demands high performance floating point computation. By using more and faster FPUs, an FPGA can provide higher data throughput.wordblocks, floating point adders and floating point multipliers are composed from non-programmable elements, resulting in amore compact block with higher speed, but less flexibility thanfine-grained logic. The main contributions are

1)A study of floating point unit architectures over a set of floating point benchmarkcircuits.

2)An analysis of the benefits of merging different types of floating point units into a larger coarse-grained floating point unit.

3) Amethodology to optimize a floating point hybrid FPGA by considering both the internal architecture of floating point units and types of floating point units incorporated into the system.

II.HYBRID FPGA

A hybrid FPGA includes coarse grained and fine-grained components, which are connected by routing tracks. Our finegrained fabric consists of an array of identical configurable logic blocks (CLBs), each containing basic logic elements (BLEs).Each BLE contains LUTs, flip flops

(FFs), support for fast carry chains,internal multiplexers and XOR gates.FPGA is an array of fine grained configurable logic blocks interconnected in a hierarchical fashion. Commercial FPGA contains coarse grained blocks such as memories and multipliers for commonly used primitives to improve efficient for specific functions. However, FPGAs are approximately 20 times larger and 4 times slower than application-specific integrated circuits (ASICs). In order to reduce this gap, considerable research has been focused on identifying more flexible and efficient coarse-grained blocks, particularly for specialized application domains such as floating point computations. It is because implementing FP operations in fine-grained FPGA consumes a large amount of resources and a number of approaches to optimize FP operations in FPGAs have been proposed. That's why floating point unit is to be implemented in a hybrid FPGA, where hybrid FPGA consists of coarse grained elements.

A. Coarse grained block

A coarse-grained floating point unit consists of FAs, FMs, and WBs. The floating point adders and floating point multipliers are double precision (64 bit) and fully IEEE754 compatible including all four rounding modes. The four rounding modes are

1) Nearest rounding, where ties round away from zero.

2) Round up, that is negative results thus round toward zero.

3) Round down, that is negative results thus round away from zero.

4) Round toward zero.

Each word block contains Nidentical bit blocks, each consisting of two 4 input look up tables and a reconfigurable register. The value of N depends on the bit-width of the FPU. Configuration bits controlling the bit blocks within the word block perform the same function.A WB can efficiently implement operations such as fixed point addition and multiplexing. So some bit level operations in FP primitives can be supported inside the FPU, to reduce the need for communication between the FPU and the fine-grained fabric.This avoids using the fine grained routing resources of the field programmable gate array for connections that can be implemented inside the FPU, see Fig. 1

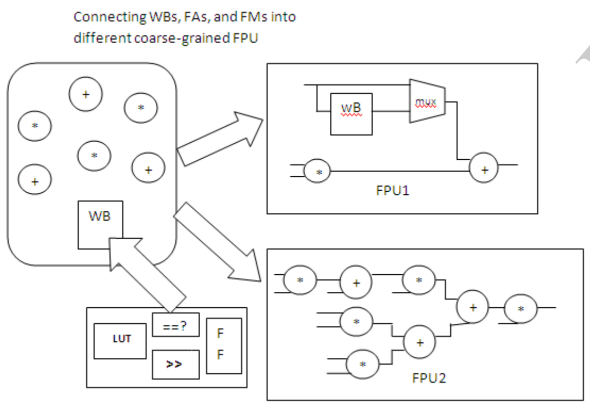

Connecting WBs, FAs, and FMs into different coarse-grained FPU
+
*
*
*
+
+
WB
WB
mux
*
+
FPU1
LUT
==?
>>
F
F
*
+
*
+
*
*
+
*
FPU2

Fig 1:Connecting WBs, FAs, and FMs into different coarse-grained FPUs.

B. Interface

Fine-grained blocks are able to connect to coarse-grained resources in various ways. Coarse grained blocks are generally large and have high I/O density, resulting in a disruption in the routing fabric, similar to that of Mega RAM blocks in the Altera.Traditional column based hybrid field programmable gate array architecture is not particularly suitable for large coarse grained blocks and concludes that coarse grained blocks should be 1) square in aspect ratio 2) closely packed together 3) positioned near the center of the field programmable gate array and 4) have I/O pins arranged on all four sides of the block.

III. CLOCK GATING

Clock gating is a very famous technique used in many circuits for reducing dynamic power dissipation. Clock gating method saves power by adding more logic to a circuit to prune the clock tree. Pruning the clock disable them portions of the circuitry so that the flip flops in them do not have to switch states. Switching states consumes power. When it is not being switched, the switching power consumption goes to zero, and only leakage currents are incurred.

Clock gating technique can be added into a design in a variety of ways. 1) Coded into the RTL code as enable conditions that can be automatically translated into clock gating logic by synthesis tools.2) The RTL designers manually inserted into the design (typically as module level clock gating) by instantiating library specific Integrated Clock Gating(ICG)cells to gate the clocks of specific modules or registers.3) Automated clock gating tools inserted into the RTL Semi-automatically. These tools either insert integrated clock gating cells into the RTL, or add enable conditions into the RTL code.

IV FPU OPTIMIZATIONS AND METHODOLOGY

In this section there are three types of optimizations: the internal structure of the floating point unit, optimizations for density and flexibility, and the merging of FPUs into larger composite structures.The techniques described here are not restricted to our specific architecture. It can be extended to optimize the other FPU architecture also.

A.Internal Optimization of FPUs

FPU consists of floating point adders and multipliers (FAs and FMs) as well as fixed point coarse-grained WBs. The first optimization is the optimization of the exact number of each type of subunit within each FPU, as well as the pattern in which these subunits are connected. Fig.1 shows an example of two different potential FPU architectures with different internal structures.

By employing common subgraph extraction FPU structure can be derived. In this methodology, by analyzing a set of benchmarks systematically, and extract patterns of FP operations that commonly appear in these circuits. Fig. 2 is an example of a common subgraph of two circuits (dscg and bfly benchmarks).

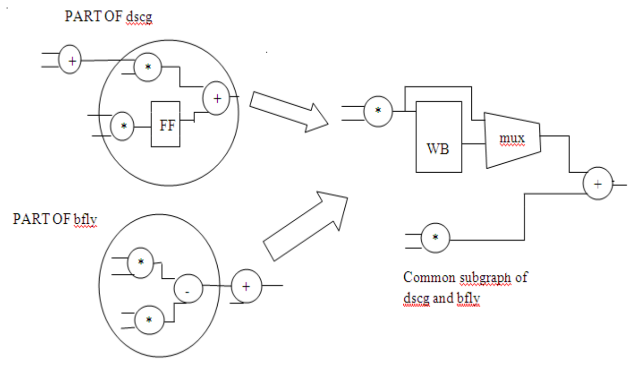


Fig 2:Common subgraph extraction for an FP applications.

A single unit which combines the common FP operations can be extracted. Thus, creation of the FPU requires considering both fixed point and floating point functions in the application circuits. In this approach it is necessary to include a multiplexer to the input of the FA or FM in the common subgraph when there is a fixed

point operationsuch as FF, XOR, and AND connecting to the floating adder or floating multiplier in one of the analyzed benchmarks. This multiplexer allows selecting internal signal fromwordblocks implementing a fixed point operation, internal signal from FM or FA implementing a floating point operation, or external signal. The combination of fixed point WBswith FAs and FBs leads to more efficient circuit implementations,which reduces the slow communication between the FPU and the fine-grained fabric.

B. System-Level Optimizations

System level optimization includes the optimization for density.Since connections between the components in the FPU can be made locally,An FPU with more computational elements achieves a greater reduction in area.However, larger blocks may require more routing resources for connections to fine-grained blocks, and may lead to a reduction in flexibilitysince it is difficult to reuse them in other applications.

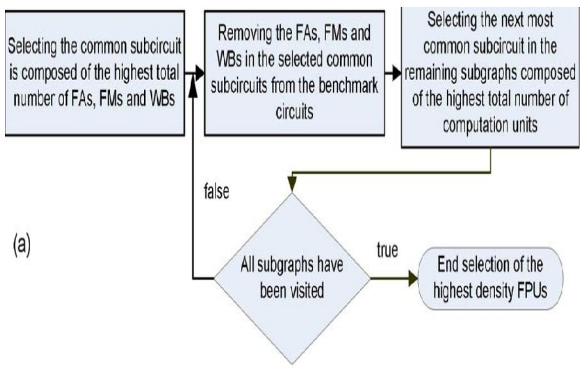

Selecting the common subcircuit is composed of the highest total number of FAs, FMs and WBs
Removing the FAs, FMs and WBs in the selected common subcircuits from the benchmark circuits
Selecting the next most common subcircuit in the remaining subgraphs composed of the highest total number of computation units
false
(a)
true
All subgraphs have been visited
End selection of the highest density FPUs

Fig 3:Flow of the selection of the highest density FPU

C.Optimization by Merging FPUs

The final type of optimization is the merging of two different types of FPUs into one composite FPU. The distinct types of FPUs exist in an FPGA, the more placementconstraints the architecture imposes, since each subgraph in the circuit may only be able to be implemented in one of the FPU types. By combining FPUs into larger FPUs, these placement constraints may be relaxed, leading to efficient implementations. Fig. 4 shows an example of merging graph15 and graph26 into a larger composite FPU

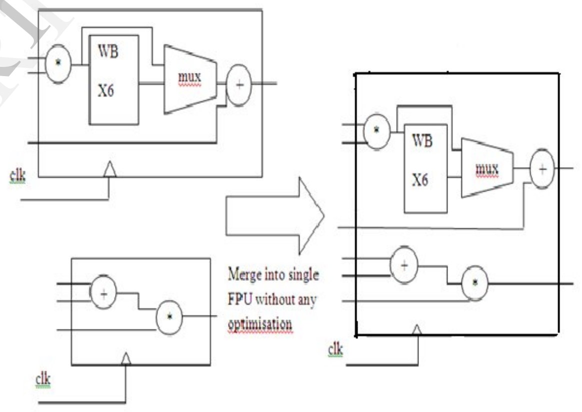
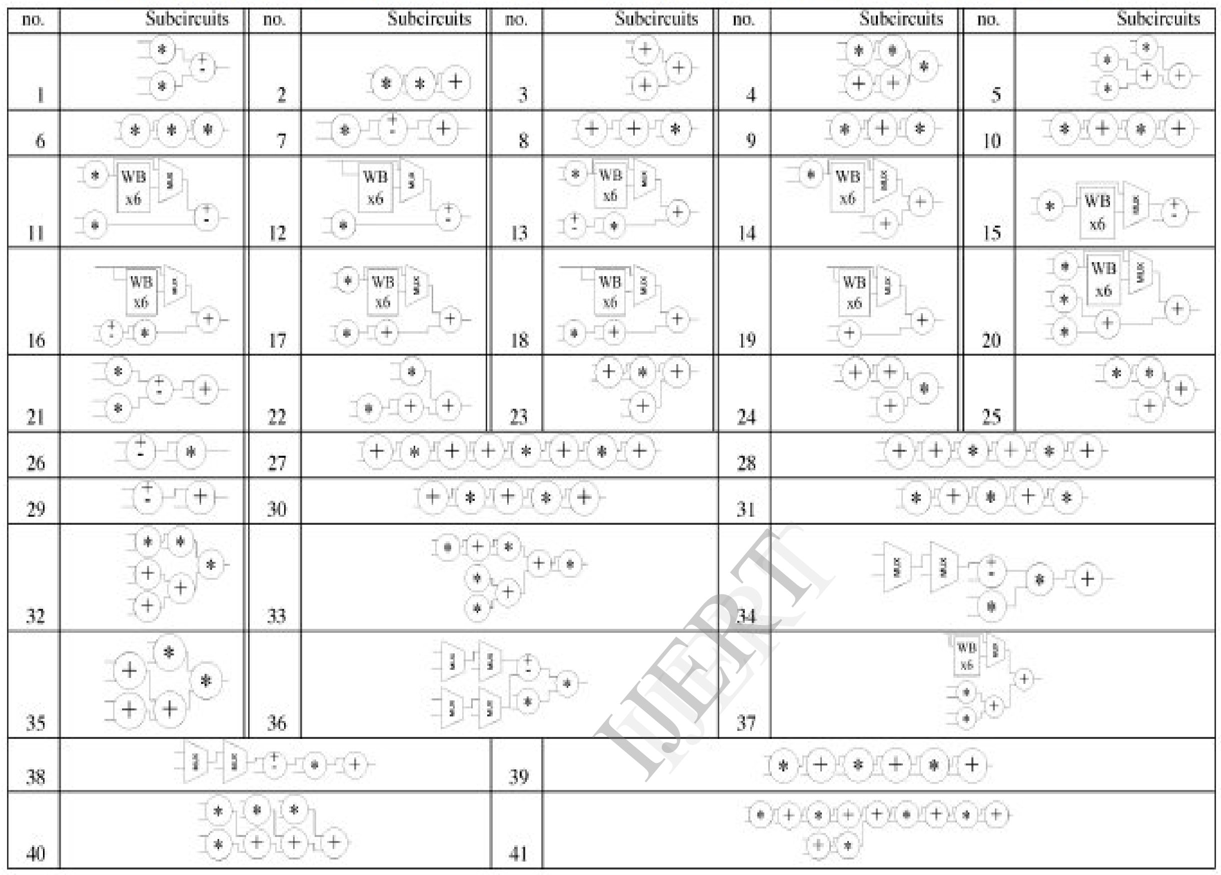


Fig 4:Merging graph15 and graph26 into a larger FPU

Two important considerations are there while merging FPUs. First, merging the different floating point units into a composite floating point unit may lead to reduced placement constraints and also leading to a reduction in the overall wirelength of the implemented circuit. The position of various FPUs is fixed so only during the placement stage the same FPU type can be swapped. This leads to inflexible placement and may introduce long wires between FPUs.

Merging different FPU types into a larger FPU may lead to a better placement and reduce the wirelength.Second, a larger FPU will require more chip area and may lead to an increase in overall wirelength.

TABLE I
COMMON SUBGRAPH STRUCTURE OCCURRED IN BENCHMARK CIRCUITS (WBX6 MEANS THERE ARE SIX WBS CONNECTED SERIALLY)



V. EVAUATION

FP benchmark circuits are introduced in this section and it evaluates the area and delay impact of internal and system-level optimizations of coarse-grained FPUs based on common subgraphs and finally optimizes the systems by merging different FPUs into a larger composite FPU.

A. FP Benchmark Circuits

To explore the design of a hybrid FPGA based on common subgraph extraction and synthesis, a set of FP designs are used as benchmark circuits.They are: 1) a data path of four digital sine-cosine generators - dscg 2) the basic computation of fast Fourier transform z = x + y where x and y are the inputs and z is the output. Both are complex numbers-bfly 3) four 4-tap finite impulse response filters- fir 4) four circuits to solve ordinary differential equations- ode 5) four 3 x3 matrix multipliers - mm3 6) a circuit to compute Monte Carlo simulations of interest rate model derivatives-bgm 7) a circuit containing 5 FAs and 4 FMs-syn2 and 8) a circuit containing 25 FAs and 25 FMs-syn7and are two synthetic benchmark circuits generated by a synthetic benchmark circuit generator. These eight double precision floating point benchmark circuits are not efficiently implemented in fine-grained FPGAs.It is only efficient in coarse grained elements.

VI. CONCLUSION

This paper presented a methodology to determine optimized coarse grained FPUs in hybrid FPGAs based on common subgraph extraction. The internal and system-level optimization of the FPU is explored. Low power can also be achieved in the FPU by using clock gating method. Finally in the FPU architecture 1) the speed of the system is the highest for implementations involving only FAs.2) higher density subgraphs produce greater reduction on area.3) Merging of FPUs can improve the speed of hybrid FPGAs.

REFERENCES

[1] I. Kuon and J. Rose "Measuring the Gap Between FPGAs and ASICs" IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 26.

[2] M. J. Beauchamp, S. Hauck, K. D. Underwood, and K. S. Hemmert, "Architectural Modifications to Enhance the Floating-Point Performance of FPGAs," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. vol. 16 Feb. 2008.

[3] C. H. Ho, C. W. Yu, P. H. W. Leong, W. Luk, and S. J. E. Wilton, "Floating-point FPGA: Architecture and modeling," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 2, pp. 1709–1718, Dec.

2009.

[4] S. S. Demirsoy and M. Langhammer, "Cholesky decomposition using fused datapath synthesis," in Proc. FPGA, 2009, pp. 241–244.

[5] M. Langhammer and T. VanCourt, "FPGA floating point datapath compiler," in Proc. FCCM, 2009, pp. 259–262.

[6] G. Govindu, L. Zhuo, S. Choi, and V. Prasanna, "Analysis of high-performance floating-point arithmetic on FPGAs," in Proc. Parallel Distrib. Process.Symp., 2004, pp. 149–156.

[7] Y. J. Chong and S. Parameswaran, "Flexible multi-mode embedded floating-point unit for field programmable gate arrays," in Proc. FPGA, 2009, pp. 171–180.

[8] A. M. Smith, G. A. Constantinides, and P. Y. K. Cheung, "Fused-arithmetic unit generation for reconfigurable devices using common subgraph extraction," in Proc. ICFPT, 2007, pp. 105–112.

[9] C. W. Yu, A. M. Smith, W. Luk, P. H. W. Leong, and S. J. E. Wilton,"Optimizing coarse-grained units in floating point hybrid FPGA," in Proc. ICFPT, 2008, pp. 57–64.