Resource Allocation for Hardware Implementations of Map

Richard Townsend
Martha A. Kim    Stephen A. Edwards

Columbia University

ASBD Workshop, June 15, 2014
Functional Programs to Functional Hardware

Functional program (Haskell) -> Compiler -> Circuit (Verilog)
Map $f \ [x_1, x_2, \ldots, x_n]$
Map $f [x_1, x_2, \ldots, x_n]$
Functional Map vs. MapReduce
Functional Map vs. MapReduce
Functional Map vs. MapReduce

Ordered
Functional Map vs. MapReduce

(Ordered)

(0, 0) (1, 3) (2, 3) (3, 3)
(0, 2) (1, 2) (2, 2) (3, 2)
(0, 1) (1, 1) (2, 1) (3, 1)
(0, 0) (1, 0) (2, 0) (3, 0)

(Unordered)
Structure of a Hardware Implementation

\[ f \]
Structure of a Hardware Implementation
Structure of a Hardware Implementation
Structure of a Hardware Implementation
Structure of a Hardware Implementation
Structure of a Hardware Implementation
Structure of a Hardware Implementation
Structure of a Hardware Implementation

\[ f(x) = f(x_1) \times x_2 \times x_4 \times x_5 \times f(x_3) \times x_6 \times x_7 \times x_8 \times x_9 \times x_{10} \]
Structure of a Hardware Implementation

\[ f(x_{10})x_{9}x_{8} \rightarrow x_{6} \rightarrow x_{2} \rightarrow x_{7} \rightarrow f(x_{3}) \rightarrow f(x_{1}) \rightarrow x_{4} \rightarrow f(x_{1}) \rightarrow x_{5} \]
Structure of a Hardware Implementation
Structure of a Hardware Implementation

\[ f(x_1) = f(x_7) \cdot f(x_3) \cdot f(x_2) \cdot f(x_6) \cdot x_8 \cdot x_9 \cdot x_{10} \]
Structure of a Hardware Implementation

\[ f(x_3) \]

\[ f(x_7) \]

\[ f(x_1) \]

\[ x_2 \]

\[ x_4 \]

\[ x_5 \]

\[ x_{10}, x_9, x_8 \]
Structure of a Hardware Implementation

still processing

stuck in buffer

holding and stalling
Structure of a Hardware Implementation
Structure of a Hardware Implementation

More Functional Units (Parallelism)

More Buffers (Utilization)
Multiple Possible Configurations...Which to Choose?

Area = 15
Multiple Possible Configurations...Which to Choose?

Area = 15

Buffers 50% size of func. unit
Multiple Possible Configurations...Which to Choose?

Area = 15

15 Functional Units
Multiple Possible Configurations...Which to Choose?

Area = 15

28 Buffers
Multiple Possible Configurations...Which to Choose?

Area = 15

$20 \times \frac{1}{2} = 10$
Workload Structure

- Best-case
- Average-case
- Worst-case
Workload Structure

Best-case

[Diagram of workload structure with best-case scenario]
Workload Structure

Best-case

Time

Average-case

Worst-case
Workload Structure

Best-case

Average-case

?
Workload Structure

Best-case

Average-case

Worst-case
Workload Structure

Best-case

Average-case

Worst-case
Workload Structure

Best-case

Average-case

Worst-case
Optimal Resource Allocation

Simulator Results

Buffer Slots per Functional Unit

Completion Time
Optimal Resource Allocation

Simulator Results

Maximizing Functional Units

2x speedup
Optimal Resource Allocation

Simulator Results

3x speedup

Maximizing Buffers
Optimal Resource Allocation

Completion Time vs. Buffer Slots per Functional Unit
Optimal Resource Allocation

Fewer Buffers => Slower
Why Are There Multiple Optima?
Why Are There Multiple Optima?
Why Are There Multiple Optima?
Performance Scales with Area

Completion Time vs. Total Area

- Ideal
- Minimum
Performance Scales with Area

- Completion Time
- Buffers Slots / Functional Unit
- Functional Units
- Total Area
Performance Scales with Area

Completion Time vs. Total Area

Buffers Slots / Functional Unit vs. Total Area

Functional Units vs. Total Area
Conclusions

Area allocation is important...
Conclusions

Area allocation is important...

...and non-obvious
Area allocation is important...

...and non-obvious

Model helps explore design space
Conclusions

Area allocation is important...

...and non-obvious

Model helps explore design space

Synthesize Efficient Hardware Implementation of Map
Conclusions

Area allocation is important...

...and non-obvious

Model helps explore design space

Synthesize Efficient Hardware Implementation of Map

Enhance our abstraction