Converging Design Flows

the same synthesis method may be used for mapping an algorithm onto both:

rDPA [Kress, 1995],
and DPA [Broderson, 2000]:

terms:
DPU: datapath unit
DPA: data path array
rDPU: reconfigurable DPU
rDPA: reconfigurable DPA

this synthesis method is a generalization of systolic array synthesis: super systolic synthesis

Time to space migration of algorithms

Problems in time to space migration of algorithms

Some have moderate interconnect requirements
Many DSP algorithms require just a pipeline
Some algorithms require excessive interconnect
Example: the Viterbi algorithm
A comprehensive taxonomy of algorithms is missing

IC interconnect: metal layers

Foundries offer up to 9 metal layers
and up to 3 poly layers
Reconfigurable interconnect fabric laid out over the rDPU cell

KressArray Family generic Fabrics: a few examples

Select mode, number, width of NNports
select Nearest Neighbour (NN) Interconnect: an example
Examples of 2 Level Interconnect: layered over rDPU cell - no separate routing areas!
more NNports: rich Rout Resources

>> Time to space migration <<

• Time to space migration
• Flowware languages
• Data Sequencers
• Sequencing through 2-D memory
• MoM architecture
• Acceleration mechanisms
KressArray Xplorer

Application
Editor
Compiler

Improvement Proposal Generator
Architecture
Improv.
Sug.
stat.
Inter.
Delay Estim.

Architecture Estimator

Compiler
ALE
Compiler
ALE

ALE-X
Compiler

Power Estimator

Power Data
HDL
Generator
Simulator

VHDL
Verilog

Data
Generator

User
ALEX
Code

Gener.
Inference
Engine (FOX)

Suggestion
Selection

User Interface

interm. form 3

Mapper

Design Rules
Generator

Datapath Generator

Map

Schedule

Scheduler

route thru only

http://kressarray.de
array size: 10 x 16 = 160 rDPUs

http://kressarray.de

Communication resource editor panel of the Xplorer user interface

Elements of the Xplorer mapping editor:

a) Routing editor panel
Elements of the Xplorer mapping editor:
b) Input port editor panel

Xplorer: Improvement Proposal Generator

Suggestions for Datapath 1 / Version 0

Xplorer: conditional swap operator

Xplorer: Macro cells

FPGA-Style Mapping for coarse grain reconfigurable arrays

Compiler
Mapper
Scheduler

KressArray DPSS
(Degath Synthesis System)

Ulrich Nageldinger
Dissertation

• ... on mapping applications onto KessArrays
• ... simultaneous routing and placement by simulated annealing
• Supporting a huge family of KessArrays
• Fuzzy logic improvement proposal generator
• profiling
• Design space exploration
**MoPL**

- **Grammar 1 (14):** 1. Program Def.
- **Grammar 2 (14):** 2. Boundary Decl's

---

**Flowware languages**

- Time to space migration
- Flowware languages
- Data Sequencers
- Sequencing through 2-D memory
- MoM architecture
- Acceleration mechanisms

---

**MoPL Subroutine**

- **Declaration Part**
  - **Ident**
  - **Array Declaration**
  - **SW Declaration**
  - **Boundary Declaration**
  - **RDLU**

---

**MoPL-3 Grammar**

- The MoPL-3 Grammar of ... of ...
  - the Map-oriented Programming Language version 3 (MoPL-3), a data-procedural programming language
  - to specify functions and operators to be mapped onto a DataPath Array (DPA) or other pipe network (hardwired as well as reconfigurable)
  - and to procedurally program data streams associated with these functions or operators

---

**Similar Programming Language Paradigms**

<table>
<thead>
<tr>
<th>Language category</th>
<th>Computer Languages</th>
<th>Xputer Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>both deterministic</td>
<td>procedural sequencing: traceable, checkpointable</td>
<td></td>
</tr>
</tbody>
</table>

Sequencing driven by:

- read next instruction, get (to instruction addr.), jump (to instruction addr.), instruction loop nesting
- no parallel loops, instruction loop escapes, instruction stream branching

---

**MoPL Example**

- JPEG zigzag scan pattern
- Flowware language example (MoPL)

---

**Scan Window Declarations**

- Compound Window Declaration
- SW Group Name
- SW Group Name
- Window Spec
- Window Names
- Window Size
- Data Type
- Data Counter
- Data Address (hardwired as well as reconfigurable)

---

**MoPL Subroutine**

- **Declaration Part**
  - **Ident**
  - **Array Declaration**
  - **SW Declaration**
  - **Boundary Declaration**
  - **RDLU**
4. rALU Set-up Declarations

- **rALU Config**
  - Structural Part
  - Top Structure
  - rALU Name

- **Do Structure**
  - While Structure

- **Sub Structure**

- **Top Structure**
  - rALU subnet is resident

- **rALU Activation**

- **Local Branch Flag**

5. Scan Pattern Declarations

- **Compound Scan Pattern Decl**
  - Pattern Name is Scan Action
  - rALU subnet Flag dependent on localBranchFlag

- **Simple Pattern Decl**
  - Pattern Name is Scan Action
  - rALU subnet Flag
  - Local Branch Flag

6. Scan Statement Declarations

- **Scan Statement Block**
  - Scan Pattern Name
  - Scan Window Name
  - Array Name

- **Scan Pattern Call**
  - Scan Pattern Name, [ ], parbegin, parend

7. Scan Pattern Call

- **Scan Statement**
  - Scan Pattern Name
  - Scan Window Name

8. Scan Statement

- **Scan Pattern Call**
  - Scan Pattern Name
  - Scan Window Name
  - Array Name

9. Scan Pattern Call
MoPL grammar 9 (14): 7. Scan Actions

7. Scan Action Declarations

Scan Action

Pattern Spec

Scan Pattern Sequence

Simple Scan

Library Scan

MoPL grammar 10 (14)

MoPL grammar 11 (14)

MoPL grammar 12 (14): 8. Expressions

8. Expression Declarations

Assignment

Expression

Simple Expression

Term

Rel Op

Factor

SW Variable

MoPL grammar 13: 9. Lexical Declarations

9. Lexical Declarations

Letter

Digit

Underscore

Scale Factor


10. Common Production Rules

Decl-Size

Range

Name-List

Data Type
**Data Sequencers**

- Time to space migration
- Flowware languages
- Data Sequencers
- Sequencing through 2-D memory
- MoM architecture
- Acceleration mechanisms

**application-specific distributed memory**

- Application-specific memory: rapidly growing markets:
  - IP cores
  - Module generators
  - EDA environments
- Optimization of memory bandwidth for application-specific distributed memory
- Power and area optimization as a further benefit
- Key issues of address generators will be discussed

*) see books by Francky Catthoor et al.

**Significance of Address Generators**

- Address generators have the potential to reduce computation time significantly.
- In a grid-based design rule check a speed-up of more than 2000 has been achieved, compared to a VAX-11/7850.
- Dedicated address generators contributed a factor of 10 - avoiding memory cycles for address computation overhead.

**Smart Address Generators**

- 1984: The GAG (generic address generator)
- 1989: Application-specific Address Generator (ASAG)
- 1990: The slider method: GAG of the MoM-2 machine
- 1991: The AGU
- 1994: The GAG of the MoM-3 machine
- 1997: The Texas Instruments TMS320C54x DSP
- 1997: Intersil HSP45240 Address Sequencer
- 1999: Adopt (IMEC)

**Distributed Memory**

- SA: scrambling and descrambling the data?
- Just in time: a new research area:
  Application-specific distributed memory:
  e. g. book by F. Catthoor et al. ...
  Data address generators - 20 years research:

For more details on Adopt see paper in proceedings CD-ROM
• Time to space migration
• Flowware languages
• Data Sequencers
• Sequencing through 2-D memory
• MoM architecture
• Acceleration mechanisms

MoM architecture: 2-D memory space, adj. scan window
grid-based design
rule check example
speed-up: >1000
complex boolean expressions in 1 clock cycle
address computation overhead: 94 %

2-D Generic Data Sequence Examples

Size adjustable at run time
square or rectangular shape
location's individual access mode: R, W, R/W, no-op
by no-op placements any wild window shape
avoid multiple read/multiple write for overlapping successive scan window positions
GAG Slider Model

Generic Address Generator

scan patterns example for illustrations of the slider model.

TU Kaiserslautern

© 2004, reiner@hartenstein.de

http://hartenstein.de

GAG Slider Operation Demo Example

GAG = Generic Address Generator

Limit Stepper

Address Stepper

Base Slider

GAG: Address Stepper

GAG = Generic Address Generator

Limit

Address

Base

limit

all 3 are copies of the same BSU stepper circuit

TU Kaiserslautern

© 2004, reiner@hartenstein.de

http://hartenstein.de

Generic Sequence Examples

video scan

-90° rotated video scan

-45° rotated (mirx (v scan))

sheared video scan

non-rectangular video scan

zigzag video scan

spiral scan

feed-back-driven scans

perfect shuffle

published in 1990

GAG Slider Operation Demo Example

© 2004, reiner@hartenstein.de

http://hartenstein.de
GAG Complex Sequencer Implementation

XMDS Scan Pattern Editor GUI

>> Acceleration mechanisms <<

Linear Filter Application

Scanline unrolling

90° Rotation of Scan Pattern
Linear Filter Application

Parallelized Merged Buffer Linear Filter Application with example image of x=22 by y=11 pixel

Speedup by MoM

Multiple scan windows

MoM anti machine an Xputer architecture

MoM distributed memory interface

example 4x4 scan windows

16 point CGFFT: mapped onto 2-D memory space

CGFFT: Nested and Parallel Scan Pattern

CGFFT: Parallel Scan Pattern Animation

MoM anti machine architecture

Speedup by MoM

Multiple scan windows

MoM anti machine an Xputer architecture

MoM distributed memory interface

example 4x4 scan windows

16 point CGFFT: mapped onto 2-D memory space

CGFFT: Nested and Parallel Scan Pattern

CGFFT: Parallel Scan Pattern Animation
CGFFT: Parallel Scan Pattern Animation

4 MAC units in parallel
in / count empty
3 MAC units in parallel

18 steps

CGFFT: Nested and Parallel Scan Pattern

outer loop scan pattern
HLScan is 3 steps [2, 0]
SP1 in 7 steps [0, 2]
goto
SP23 in 7 steps [0, 1]
3 in parallel
inner loop compound scan patterns

>> Acceleration mechanisms <<

• Time to space migration
• Flowware languages
• Data Sequencers
• Sequencing through 2-D memory
• MoM architecture
• Acceleration mechanisms

Speed-up Enablers

Hier eine Liste
DRC 4 orders of magnitude
Address computation overhead
Translate into super-systolic rather than into instruction streams
Determine interconnect fabrics by compilation, but not before fabrication
Determine memory architecture by compilation, but not before fabrication

Acceleration Mechanisms

• parallelism by multi bank memory architecture
• auxiliary hardware for address calculation
• address calculation before run time
• avoiding multiple accesses to the same data.
• avoiding memory cycles for address computation
• improve parallelism by storage scheme transformations
• improve parallelism by memory architecture transformations
• alleviate interconnect overhead (delay, power and area)