Refinements in Data Manipulation Method for Coarse Grained Reconfigurable Architectures

Takuya Kojima and Hideharu Amano
Keio University, Japan

14th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC 2019)
Importance of Programmability and High Energy Efficiency

- Forthcoming
  - IoT devices
  - Wearable computers
  - Edge computing

- Challenges for these devices
  - Programmability
    - To satisfy various demands
  - High energy efficiency
    - To extends long battery life
CGRAs: Coarse-Grained Reconfigurable Architectures

- CGRAs
  - Support word-level reconfiguration (↔ bit-level of FPGAs)
  - Have many PEs (Processing Element) in 2D grid
  - Change functionality for each ALU & interconnection between PEs dynamically or statically
Power-hungry Dynamic Reconfiguration

- **Dynamic Reconfiguration**
  - Changes configuration cycle-by-cycle
  - Provides more flexibility
  - Causes large dynamic power consumption

Details of power consumption for a dynamic reconfiguration CGRA[1]

---

SF-CGRAs: Straight-Forward CGRAs

Key features of straight-forward CGRAs
- Limited data flow direction
- Less frequent reconfiguration
- Pipelined PE array
- High energy efficiency


VPCMA: Variable Pipelined Cool Mega Array [2]

- PE array consists of
  - 8 x 12 PEs
  - 7 pipeline registers

- PE has
  - No Register file
  - No clock tree

- Pipeline register works in
  1. latch mode or
  2. bypass mode

- μ-Controller
  - Controls data transfer
    data mem. ↔ PE array

Computation on the PE array

- Fetch registers are connected to input of the PE array
- Gather registers are connected to output of the PE array
- The micro-controller
  - Writes data to the fetch registers
  - Read result from the gather registers
Computation on the PE array

- Fetch registers are connected to input of the PE array
- Gather registers are connected to output of the PE array
- The micro-controller
  - Writes data to the fetch registers
  - Read result from the gather registers
Variable Pipeline Structure

- No registers in each pipeline stage
  → Pure combinational circuit
- Clock tree only for activated pipeline registers
- Variable pipeline structure depending on application
Multi-cycle Execution on PE Array

Micro-controller

- A custom tiny RISC processor controls the processing
- "Fetch" op kicks off the execution
- "Gather" op writes back the results
- "Delay" op specifies delay time of "Gather" execution
- "Branch" op makes a loop
Multi-cycle Execution on PE Array

- **Delay**
- **Fetch**
  - stage1
  - stage2
  - stage3
  - stage4
  - Gather
- **NOP**
- **Delay**
  - 8 cycles

To adjust the timing by inserting other instr.

- **Micro-controller**
  - A custom tiny RISC processor controls the processing
  - "Fetch" op kicks off the execution
  - "Gather" op writes back the results
  - "Delay" op specifies delay time of "Gather" execution
  - "Branch" op makes a loop

Fused into an instruction
Data Manipulator of VPCMA

- Data manipulator
  - Placed between Dmem & PE array
  - Transfers any input data to any outputs
  - Loads at most consecutive 12 data from 12 mem banks
  - Increments addr. automatically for next fetch
Data Manipulator of VPCMA

- Data manipulator
  - Placed between Dmem & PE array
  - Transfers any input data to any outputs
  - Loads at most consecutive 12 data from 12 mem banks
  - Increments addr. automatically for next fetch

---

**Transfer Table #0**

<table>
<thead>
<tr>
<th>dst.</th>
<th>src.</th>
<th>mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>col0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>col1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>col2</td>
<td>N/A</td>
<td>0</td>
</tr>
<tr>
<td>col3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>col4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>col5</td>
<td>N/A</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Ultra Low Power Consumption of CMA

- No-Pipelined version of CMA[6]
  - Works with Lemon battery
  - Achieves 743 MOPS/mW (297MOPS/0.4mW)

- VPCMA
  - Keeps the same energy efficiency
  - Achieves 4x higher peak performance

- Problem
  - Less flexibility because of saving too much energy

Limitation of data handling in VPCMA

for $i \leftarrow 0$ to $N = 64$ do
  $c[i] \leftarrow a[i] \times b[i \mod 16]$ 
end for

Loop example

- Data manipulator cannot access multiple data more than 12 step distance simultaneously
- needs data rearrangement
- often incurs extra copy of data

Memory allocation in bank memory
Data manipulator cannot access multiple data more than 12 distance simultaneously → needs data rearrangement → often incurs extra copy of data
Other limitations of VPCMA

Also, VPCMA

1. Suffers from a lack of constant registers for the PE array
   - A PE row (12 PEs) share two const regs. or borrows from other rows via interconnection
   - Degrades mappability of complex kernels

2. Depends on a host processor for overall control
   - Micro-controller basically controls data transfer & loop counter
   - All of other controls (e.g. reconfiguration) are carried out by the host processor even if trivial change is needed
A new architecture VPCMA2

- Relaxing aforementioned limitations
  1. Improved bank access by new data manipulator
  2. Refined connectivity of constant registers
     - PE array has 16 constant registers (same as VPCMA)
     - All PE can use any 16 registers
  3. Introduced an extended data bus for micro-controller
New Data Manipulator

Offset values for each bank is introduced
Relaxed the limitation of consecutive data access
New Data Manipulator

Offset values for each bank is introduced
Relaxed the limitation of consecutive data access
Micro-controller can handle any data in other modules
**Evaluation Setup**

- An implementation of VPCMA2
  - Using Renesas SOTB 65-nm technology
    - LSTP (Low STanby Power) version
  - Synthesized by Synopsys Design Compiler 2017

- A real chip of VPCMA[7]
  - Fabricated same technology
    - LP (Low Power) version (75% slower than LSTP)

Hardware overhead

<table>
<thead>
<tr>
<th></th>
<th>VPCMA [7]</th>
<th>VPCMA2 1-cycle f/g</th>
<th>VPCMA2 2-cycle f/g</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Frequency (MHz) (75% scaled)</td>
<td>87.71</td>
<td>95.23 (71.42)</td>
<td>125.0 (93.75)</td>
</tr>
<tr>
<td>Cell Area (mm²) without PE array</td>
<td>10.04</td>
<td>14.55</td>
<td>14.22</td>
</tr>
</tbody>
</table>

- Improved data manipulator could increase critical path delay (i.e. degradation of operating freq.)
  - 2 version of designs are evaluated
    1. Fetch&Gather are performed within 1 cycle (naïve)
    2. Fetch&Gather take 2 cycles (to divide the long critical path)
- 2-cycle f/g
  - Does not have any effects on the frequency
  - Causes 40% increase of cell area
Comparison of Power Consumption

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Voltage</td>
<td>0.55 V</td>
<td>0.75 V</td>
</tr>
<tr>
<td>Static Power</td>
<td>0.126 mW</td>
<td>0.0252 mW</td>
</tr>
<tr>
<td>Dynamic Power</td>
<td>3.337 mW</td>
<td>4.029 mW</td>
</tr>
<tr>
<td>Total Power</td>
<td>3.463 mW</td>
<td>4.053 mW</td>
</tr>
</tbody>
</table>

17% increase

Compared to VPCMA(real chip), VPCMA2 (simulation)

- Reduces static power consumption because of process difference (not architecture difference)
- Increases dynamic power consumption because of the improved functionality and partially due to the higher standard voltage
Enhanced Application Mappability

![Mapping result of DCT by Genetic algorithm-based mapper][6]

- VPCMA2 can accommodate large & complex kernel but VPCMA cannot
Improved both of mappability & bank access contribute to 1.46x higher performance
Conclusion

- This work points out a problem by data handling limitations of VPCMA
- A new SF-CGRA: VPCMA2 is proposed to relax the limitations
- Evaluation results shows
  - 10% area overhead (as a whole of chip)
  - No degradation of operating frequency
  - 17% power overhead
  - 46% performance improvement
- Future work
  - Analysis of effectiveness for other architectures
  - Evaluation of real chip implementation (under fabrication)
End of presentation
Thank you for your attention

Any questions?