An Leakage Minimization for Partially Dynamically Reconfigurable FPGAs

Jen-Wei Hsieh, Yuan-Hao Chang, and Wei-Li Lee
Outline

• Introduction
  – Field-Programmable Gate Array
  – Pre-fetch scheme & Leakage Waste

• Motivations

• Enhanced Leakage Aware Algorithm

• Performance Evaluation

• Conclusion
Introduction

• Field-Programmable Gate Array (FPGA)
  – Economy
  – Fast to market

• Runtime reconfiguration capability in FPGA
  – Allowed to reconfigure different tasks when another task is still running.
  – Tasks are separated to reconfigure(R) and execution component(E).
Introduction

- **Dependency relations**
  - 1 → 2 → 3 → 4

- **Pre-fetch scheme**
  - Shorten the schedule length.

- **Leakage waste (LW)**
  - Power consumption between the R component and E component
Outline

• Introduction
• Motivations
  – Power saving
  – Disadvantages of related work
• Enhanced Leakage Aware Algorithm
• Performance Evaluation
• Conclusion
Motivations

- Power issues
  - 0.13um -> 60nm -> 45nm -> 32nm
  - Ignore power consumption at 0.13um.

- Post-placement Leakage-aware task Algorithm (PLA) [1]

Example of PLA

(c) PLA result placement
Disadvantages of PLA

• Initial placement dependency
Disadvantages of PLA

- ALAP causes task jam
- Increase schedule length
Outline

• Introduction
• Motivations
• Enhanced Leakage Aware Algorithm
  – Architecture and Parameters
  – Priority Dispatcher
  – Split-aware Placement
• Performance Evaluation
• Conclusion
Architecture

• Target FPGA system model [1]

• ELAA architecture
  – Priority Dispatcher
  – Split-aware Placement
Parameters

- Task parameters in ELAA
  - Task $T : \{ ET, RT, W, D, dep \}$

- Output parameters of ELAA
  - $P : \text{priority of task}$
  - $et (rt) : \text{begin time of } E (R)$
  - $dr : \text{relations between tasks}$
  - $S(dr) : \text{group slack of } dr$
Task Info Table and Relation Graph

<table>
<thead>
<tr>
<th>Taski</th>
<th>RT_i</th>
<th>ET_i</th>
<th>W_i</th>
<th>dep</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>null</td>
</tr>
<tr>
<td>T_2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>T_3</td>
</tr>
<tr>
<td>T_3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>null</td>
</tr>
<tr>
<td>T_4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>T_1</td>
</tr>
<tr>
<td>T_5</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>null</td>
</tr>
<tr>
<td>T_6</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>T_6</td>
</tr>
<tr>
<td>T_7</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>null</td>
</tr>
<tr>
<td>T_8</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>T_2, T_7</td>
</tr>
</tbody>
</table>

Diagram:

1 → 4 → 6
3 → 2 → 7
5
8
Parameter examples

- Example of $\text{DR}$

$$\text{DR} = \{ \text{dr}_1, \text{dr}_2, \text{dr}_3, \text{dr}_4, \text{dr}_5 \}$$

- $\text{dr}_1 : (T_1, T_4)$
- $\text{dr}_2 : (T_1, T_5)$
- $\text{dr}_3 : (T_2, T_4)$
- $\text{dr}_4 : (T_2, T_5)$
- $\text{dr}_5 : (T_3, T_4)$
Parameter examples

- Example of $S(\text{dr})$

\[ S(\text{dr4}) = 10 - (1 + 3 + 4) = 2 \]

R2 in row 0, $S(\text{dr4}) + 1 = 3$

\[ S(\text{dr5}) = 10 - (1 + 5 + 3) = 1 \]

R3 in row 0, $S(\text{dr5}) + 1 = 2$

\[
S(dr_i) = D_{\text{group}} - \left\{ RT_{\text{head}} + \left( \sum ET_k \right) \right\}, \forall RT_k \in dr_i
\]
The rules of Priority Dispatcher

**Rule 1**: The least $S(dr)$

**Rule 2**: The largest $W$

**Rule 3**: The largest $ET$

**Rule 4**:
- a: Check dependency relations
- b: Put to block queue

**Rule 5**: Randomly assign
Split aware check

- **AvailableRow**: The remain available row of system.
- **RowRequest**: The total row request of remain tasks.

\[
\text{AvailableRow: } TRow - CRow + 1
\]
\[
\text{RowRequest: } LeftRT + LastET
\]
Example of split situation

- Execution part
- Leakage waste
- Reconfigure part

AvailableRow

RequestRow
Result of split situation

<table>
<thead>
<tr>
<th>: Execution part</th>
<th>: Leakage waste</th>
<th>: Reconfigure part</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Deadline

19/34
Step example of ELAA

- Calculate the group slack $S(dr)$

**Table1 graph:**

```
   1      3      5
     \   /     /   \
    4 --- 2 --- 7
     \   /     /   \
      6     8
```

$S(dr_1) = D_10 - (1 + 5 + 3) = 1$

$S(dr_2) = D_6 - (R_{T_1} + (E_{T_1} + E_{T_6})) = 2$

$S(dr_3) = D_8 - (R_{T_3} + (E_{T_3} + E_{T_2} + E_{T_8})) = 0$

$S(dr_4) = D_8 - (R_{T_7} + (E_{T_7} + E_{T_8})) = 1$

$S(dr_5) = D_5 - (R_{T_5} + (E_{T_5})) = 6$

$DR = \{ dr_1: \{T_1, T_4\}, 
        dr_2: \{T_1, T_6\}, 
        dr_3: \{T_3, T_2, T_8\}, 
        dr_4: \{T_7, T_8\}, 
        dr_5: \{T_5\} \}$

<table>
<thead>
<tr>
<th>Taski</th>
<th>RT_i</th>
<th>ET_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>$T_2$</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$T_3$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$T_4$</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$T_5$</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>$T_6$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$T_7$</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>$T_8$</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
Step example of ELAA

- Schedule tasks

\[
S(dr_1) = 0 \quad (T1,T4) \quad ; \quad S(dr_2) = 0 \quad (T1,T6) \\
S(dr_3) = 0 \quad (T3,T2,T8) \\
S(dr_4) = 1 \quad (T7,T8) \quad ; \quad S(dr_5) = 6 \quad (T5)
\]
Comparison of ELAA and PLA

Produce an infeasible schedule
Outline

- Introduction
- Motivations
- Enhanced Leakage Aware Algorithm
  - Performance Evaluation
    - Experiment setup
    - Experiment result
- Conclusion
Experiment setup

• Schedule Simulator
  – Schedule area
    • 50 rows and control column number between 15 ~ 50 units.
  
– Generated **5000** different Task sets as input.
  • Randomly generate RT(1~5 row), ET(1~10 row).
  • Control dependency relation percents (0%~100%).
  • Control number of tasks in a task set (3~11).
Experiment setup

• Compare 3 types of scheduler with our simulator
  – Greedy
  – PLA
  – ELAA

• Performance Metrics for these 3 algorithm
  – Comparison of leakage wastes units
  – Comparison of schedulability

• Impacts of number of tasks in a task set.
• Impacts of dependency relations.
• Effects of different available column.
Impacts of the number of tasks in a task set

- Schedule area: 20x50
- Dependency: 70%
Impacts of dependency relations

- Schedule area: 20x50
- 7 tasks in a task set
Effects of available column

• 8 tasks in a task set.
• Dependency 70%.
Impacts of the number of tasks in a task set

- Schedule area: 20x50
- Dependency: 70%
Impacts of dependency relations

- Schedule area: 20x50
- 7 tasks in a task set
Effects of available column

- 8 tasks in a task set.
- Dependency 70%.
Outline

- Introduction
- Motivations
- Enhanced Leakage Aware Algorithm
- Performance Evaluation
- Conclusion
Conclusion

• Proposed an offline algorithm ELAA
  – Generate a new schedule placement.
  – More stable than other algorithm.

• The average leakage waste saving about 95%.

• Keep a low infeasible ratio (less than 5%).
Q and A