A Technology Mapping Algorithm for CPLD Architectures

TingTing Hwang and C. L. Liu
Outlines

- Target Architecture
- Motivation and Our algorithms
- Experimental Results
- Conclusions
Target Cell Architecture

- 2-level logic
- 3-constraints, \((i,p,o)\)-block
Motivation

- Average number of product term used in a LUT grows linearly with input-size $k$ [4,5,6]
  - Satisfy input-size first.
- LUT-based mapping algorithms been widely studied
  - Utilization of an efficient LUT-based algorithm.
Our Algorithm

- Two objectives
  - Area minimization
  - Delay/depth minimization
- Two stages

![Diagram of Single-output PLA blocks transforming to Multi-output PLA blocks](image-url)
Area Minimization

Technology Independent
Optimized Logic

Single-output Mapping

Selection of Inputs $k$
for LUT-mapper

Area Minimization
LUT Mapping

Calculation of the
Number of
Product Term
(ESPRESSO)

Construction of Partial
Network for Re-mapping

Partial Network $\neq \emptyset$

YES

Bin Packing

Network of PLAs

NO

Multiple-output Packing
Step 1 – Select k (Area)

- Selection of Input-size k for LUT Mapping
  - i-inputs, o-outputs, p-product terms.
  - Easy to pack the signal-output PLA blocks generated by mapping phase.
The Equation to Select $k$

- Average number of product term used in a LUT grows linearly with input-size $k$

<table>
<thead>
<tr>
<th>k-LUT</th>
<th>Avg. #ptersms</th>
<th>$o=4$</th>
<th>$p=12$</th>
<th>Avg. product term sharing ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2.88</td>
<td>10.5</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.30</td>
<td>12.0</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3.84</td>
<td>13.9</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5.12</td>
<td>18.6</td>
<td>1.1</td>
<td></td>
</tr>
</tbody>
</table>

"FPGA Performance versus Cell Granularity"

$\frac{\text{Avg number of used in a } k\text{-LUT}}{1.1} \leq p$

Product term constraint

Select k

LUT Mapping  Pterms check  Partial net.  Packing
Step 4 – Partial Network

Select k → Re-Mapping → Pterms check → Partial net. → Packing
Step 4 – Partial Network (Area)

- Construction of partial Network for Re-mapping
  - Collection all $lbp$-infeasible logic blocks into partial network.
  - $lbp$ – number of product terms constraint of LUT-block.
  - If the number of product terms of a LUT-block large than $lbp$, the LUT-block is $lbp$-infeasible and should be re-mapping.
Step 4 – $lbp$ Value vs. Area
Step 4 – \( \text{lbp} \) Value vs. Area
Step 4 – \( lbp \) Value vs. Area

Select \( k \)

LUT Mapping

Pterms check

Partial net.

Packing
Step 4 – Variable $k$

- $lp =$ p
  - All $lp$-infeasible be collected to form a partial network
  - We use a heuristic method to change $k$ for re-mapping
    - We try the same value of $k$ in first iteration,
    - If no feasible logic block produced, $k=k-1$. ...
  - Until the partial network is empty
Delay/Depth Minimization

Technology Independent
Optimized Logic

Single-output Mapping

Selection of Inputs $k$
for LUT-mapper

Delay Minimization
LUT Mapping

Calculation of the Number of Product Term (ESPRESSO)

Weighted Functional Decomposition

All $(i,p)$-feasible

NO

Multiple-output Packing

Bin Packing

Network of PLAs

YES
Step 2 – LUT-Mapping (Delay)

- Delay minimization LUT-mapping
  - Any delay minimization LUT-mapping can be applied.
  - *Flow-Map* [Cong et al. 94] is used for delay optimization in our algorithm.
  - *Flow-Map* is a depth optimized algorithm.
  - The input-size k selection method is similar to area minimization Step 2.
**Step 4 – Decomposition (Delay)**

- **Functional Decomposition**

| Infeasible block \( f \), Bound set: \( \{x_1, \ldots, x_i\} \), Free set: \( \{x_{n-i}, \ldots, x_n\} \) | Decompose | Output Encoding function \( d_i \), Base function \( g \) |

Depth increasing is detectable
Satisfy input-size constraint
Step 4 – Decomposition (Delay)

- Weighted Functional Decomposition
  - Each fanin has a value of level, the primary input is 0.
  - Sort fanins by level.

\[ level(v) = \max_{v_j \in \text{fanin}(v)} level(v_j) + 1 \]

- \( \text{level}(f) = 6 \)
- \( \text{lbp}-\text{infeasible} \)
- \( a \ b \ c \ d \ e \ h \ l \)
- \( 0 \ 5 \ 5 \ 0 \ 1 \ 2 \ 5 \)

Candidate bound sets:
- 0: \{a\}
- 0: \{a,d\}
- 1: \{a,d,e\}
- 2: \{a,d,e,h\}
- 5: \{a,d,e,h,b\}
- 5: \{a,d,e,h,b,c\}
- 5: \{a,d,e,h,b,c,l\}

Select the best one
Step 4 – Bound Set Cost

- Cost function for selection bound set

\[
\text{Cost} = \alpha \times \left( \sum_{k=1}^{j} P(d_k) + P(g) + \sum_{f \in S} P(f) \right) + (1 - \alpha) \sum_{q \in C} P(q)
\]

- \(P(g)\) = The number of product term of block or function \(g\).
- \(d_k\) = decoding functions.
- \(g\) = base function.
- \(S = \{f \mid P(f) > p, \ f \text{ is an encoding function or base function}\}\).
- \(C = \{q \mid q \text{ is decoding function with critical fanins}\}\).
Delay/Depth Minimization

Technology Independent Optimized Logic

Single-output Mapping

Selection of Inputs k for LUT-mapper

Delay minimization LUT Mapping

Calculation of the Number of Product Term (ESPResso)

Selection of Inputs k for LUT-mapper

Weighted Functional Decomposition

All (i,p)-feasible

NO

Multiple-output Packing

Bin Packing

Network of PLAs

NO
Experiment Results

- Algorithms were implemented in C.
- Subset of MCNC benchmarks is used for experiment.
- Execution with SIS framework. [E. M. Sentovice et al. 92]
- SIS LUT-mapping script is used for area minimization.
- *Flow-Map* is used for delay minimization.
- Results are compared with previous PLA mapping Tool *TEMPLA*. [J. H. Anderson, DAC 98] and LUT-mapper *Flow-Map* [Cong et al. 94].
## Experiment Result – $k$ vs. Area

- **LUT**: number of single-output LUT nodes.
- **PLA**: number of multi-output (10,12,4)-PLA.
- **UT** = LUT / PLA.

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>$k=7$</th>
<th></th>
<th></th>
<th>$k=6$</th>
<th></th>
<th></th>
<th>$k=5$</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LUT</td>
<td>UT</td>
<td>PLA</td>
<td>LUT</td>
<td>UT</td>
<td>PLA</td>
<td>LUT</td>
<td>UT</td>
<td>PLA</td>
</tr>
<tr>
<td>apex6</td>
<td>136</td>
<td>2.83</td>
<td>48</td>
<td>144</td>
<td>3.06</td>
<td>47</td>
<td>166</td>
<td>3.25</td>
<td>51</td>
</tr>
<tr>
<td>dalu</td>
<td>164</td>
<td>2.10</td>
<td>78</td>
<td>197</td>
<td>3.08</td>
<td>64</td>
<td>229</td>
<td>3.31</td>
<td>69</td>
</tr>
<tr>
<td>rot</td>
<td>150</td>
<td>2.88</td>
<td>52</td>
<td>162</td>
<td>3.38</td>
<td>48</td>
<td>179</td>
<td>3.72</td>
<td>48</td>
</tr>
<tr>
<td>x3</td>
<td>142</td>
<td>2.96</td>
<td>48</td>
<td>148</td>
<td>3.22</td>
<td>46</td>
<td>189</td>
<td>3.57</td>
<td>53</td>
</tr>
<tr>
<td>frg2</td>
<td>171</td>
<td>2.80</td>
<td>61</td>
<td>180</td>
<td>3.33</td>
<td>54</td>
<td>231</td>
<td>3.79</td>
<td>61</td>
</tr>
<tr>
<td>i7</td>
<td>188</td>
<td>1.98</td>
<td>95</td>
<td>215</td>
<td>2.53</td>
<td>85</td>
<td>252</td>
<td>2.93</td>
<td>86</td>
</tr>
<tr>
<td>pair</td>
<td>253</td>
<td>2.01</td>
<td>126</td>
<td>295</td>
<td>3.01</td>
<td>98</td>
<td>346</td>
<td>3.36</td>
<td>103</td>
</tr>
<tr>
<td>apex5</td>
<td>185</td>
<td>2.53</td>
<td>73</td>
<td>203</td>
<td>2.89</td>
<td>71</td>
<td>247</td>
<td>3.25</td>
<td>76</td>
</tr>
<tr>
<td>C499</td>
<td>88</td>
<td>2.20</td>
<td>40</td>
<td>78</td>
<td>2.11</td>
<td>37</td>
<td>82</td>
<td>1.86</td>
<td>44</td>
</tr>
<tr>
<td>duke2</td>
<td>124</td>
<td>2.38</td>
<td>52</td>
<td>144</td>
<td>3.06</td>
<td>47</td>
<td>153</td>
<td>3.40</td>
<td>45</td>
</tr>
<tr>
<td>C2670</td>
<td>133</td>
<td>2.89</td>
<td>46</td>
<td>115</td>
<td>2.74</td>
<td>42</td>
<td>142</td>
<td>3.23</td>
<td>44</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>1734</td>
<td>2.41</td>
<td>719</td>
<td>1889</td>
<td>2.96</td>
<td>639</td>
<td>2216</td>
<td>3.26</td>
<td>680</td>
</tr>
</tbody>
</table>
Area Comparisons

- TEMPLA is also implemented in the SIS environment and utilizes the ESPRESSO minimization routine.
- Input-size for LUT-mapping is 6

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>(10,12,4)-PLA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OURS</td>
</tr>
<tr>
<td>apex6</td>
<td>47/7</td>
</tr>
<tr>
<td>dalu</td>
<td>64/9</td>
</tr>
<tr>
<td>rot</td>
<td>48/11</td>
</tr>
<tr>
<td>x3</td>
<td>46/6</td>
</tr>
<tr>
<td>frg2</td>
<td>54/8</td>
</tr>
<tr>
<td>i7</td>
<td>85/10</td>
</tr>
<tr>
<td>pair</td>
<td>98/10</td>
</tr>
<tr>
<td>apex5</td>
<td>71/7</td>
</tr>
<tr>
<td>C499</td>
<td>37/6</td>
</tr>
<tr>
<td>duke2</td>
<td>47/7</td>
</tr>
<tr>
<td>C2670</td>
<td>42/9</td>
</tr>
<tr>
<td>Total</td>
<td>639/90</td>
</tr>
<tr>
<td>Ratio</td>
<td>1/1</td>
</tr>
</tbody>
</table>
Delay Comparisons

- Our algorithm is 54% improvement in delay and with 29% increase in area
- Input-size for LUT-mapping is 7

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>(10,12,4)-PLA</th>
<th>OURS</th>
<th>OURS-area</th>
<th>TEMPLA</th>
<th>Flow-Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>apex6</td>
<td>51/3</td>
<td>47/7</td>
<td>49/5</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>dalu</td>
<td>84/4</td>
<td>64/9</td>
<td>85/10</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>rot</td>
<td>69/5</td>
<td>48/11</td>
<td>47/11</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td>57/3</td>
<td>46/6</td>
<td>50/6</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>frg2</td>
<td>81/4</td>
<td>54/8</td>
<td>54/7</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>i7</td>
<td>115/4</td>
<td>85/10</td>
<td>88/7</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>pair</td>
<td>132/5</td>
<td>98/10</td>
<td>92/12</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>apex5</td>
<td>104/3</td>
<td>71/7</td>
<td>73/6</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>C499</td>
<td>60/4</td>
<td>37/6</td>
<td>39/9</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>duke2</td>
<td>48/3</td>
<td>47/7</td>
<td>45/10</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>C2670</td>
<td>78/5</td>
<td>42/9</td>
<td>52/12</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>879/43</td>
<td>630/90</td>
<td>677/95</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>Ratio</td>
<td>1/1</td>
<td>-39.5%/+52.2%</td>
<td>-29.7%/+54.7%</td>
<td>+0%</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

- We presented an efficient algorithm to map a logic network into CPLD.
- For a given PLA block architecture, we studied the input and product term constraints for LUT mapping step.
- Compare to TEMPLA
  - 6% improvement in area, 5% in delay (area minimization)
  - 54.7% improvement in delay with 29.8% area increase (delay minimization)
- Compare to Flow-Map
  - In most cases, no delay increase.