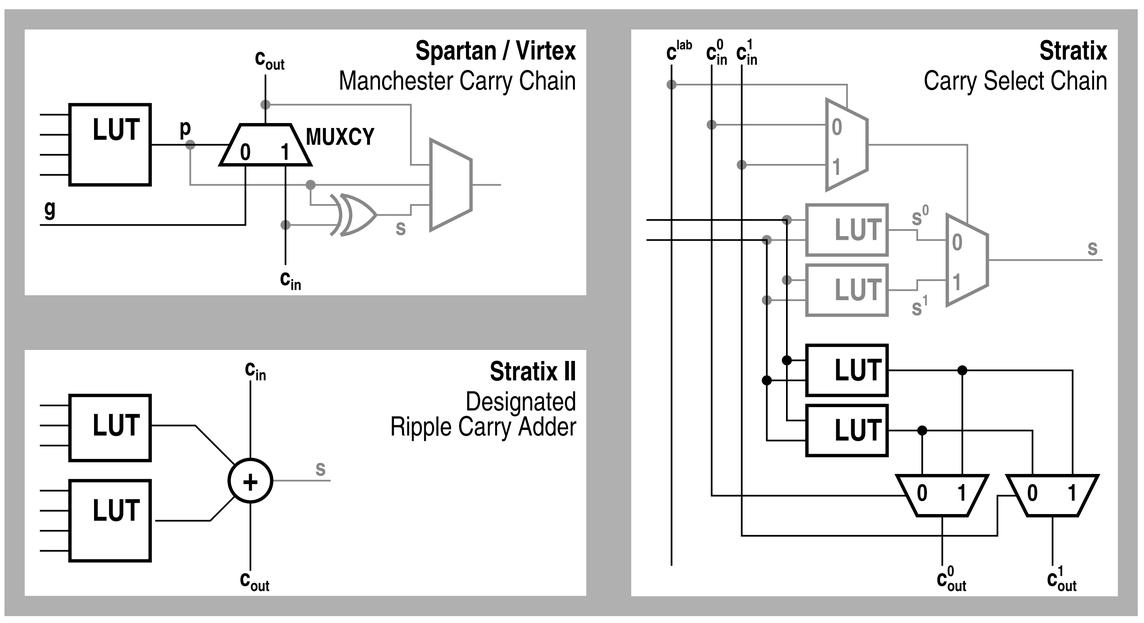
TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik Institut für Technische Informatik, Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur

MAPPING BASIC PREFIX COMPUTATIONS TO FAST CARRY-CHAIN STRUCTURES

Thomas B. Preußer, Rainer G. Spallek
Institute of Computer Engineering, Technische Universität Dresden, Germany
Email: *{preusser,rgs}@ite.inf.tu-dresden.de*

Carry Chain Structures



Prefix Computation

$\underline{x} \mapsto \underline{y}$ with:

$$\begin{aligned} y_0 &= x_0 \\ y_1 &= x_0 \circ x_1 \\ y_2 &= x_0 \circ x_1 \circ x_2 \\ &\dots \end{aligned}$$

In General:

$$y_i = \bigcirc_{j=0}^{i} x_j$$

Structural Mapping

- Automated for Stratix by Frederick and Somani (FPGA'08).
- By engineer using low-level device-specific primitives.
- By engineer using portable binary addition!
- Future Goal: **Automation!**

Mapping Procedure

Running Example: Token Propagation

$c_0 = c_{\text{in}}$
for $i \in [0, n)$:
$c_{i+1} = a_i + \overline{b_i} c_i = \overline{\overline{b_i}\overline{a_i}} \cdot a_i + \overline{b_i}\overline{a_i} \cdot c_i$
$r_i = b_i c_i$
$c_{\text{out}} = c_n$

1. **Derivation of the Carry Propagation**
   For each position in the chain, identify the three carry-propagation cases as determined by the local inputs:

Case	c_{i+1}	Description
k_i: Kill	0	No Carry / Token
p_i: Propagate	c_i	Forward Carry / Token
g_i: Generate	1	Carry / Token

   For the token propagation, we obtain:

$$\begin{aligned} p_i &= \overline{a_i}\overline{b_i} \\ g_i &= a_i \end{aligned}$$

2. **Determination of Addends**
   For mapping the function to an addition, find addends $\underline{x} = (x_{n-1}, \dots, x_1, x_0)$ and $\underline{y} = (y_{n-1}, \dots, y_1, y_0)$ that induce the desired carry propagation at the individual chain positions:

Case	x_i	y_i	
$k_i = 1$	0	0	
$p_i = 1$	$\left\{ \begin{smallmatrix} 0 \\ 1 \end{smallmatrix} \right.$	$\left. \begin{smallmatrix} 1 \\ 0 \end{smallmatrix} \right\}$	Choose Any
$g_i = 1$	1	1	

   The following straightforward choice suffices the example token propagation:

$$\begin{aligned} x_i &= g_i + p_i = a_i + \overline{b_i} \\ y_i &= g_i \qquad\quad = a_i \end{aligned}$$

3. **Fix up of Sum Bits**
   The standard binary word addition yields: $s = x + y$.
   The originally intended function needs to be derived from the generated sum bits and the local inputs. Direct access to the intermediate carry signals is not available when using the binary word addition. They can, however, be derived from the local p_i inputs and s_i outputs as $s_i = p_i \oplus c_i$.
   Granting the token to the first requestor receiving it via the carry chain, we obtain for our example:

$$r_i = b_i c_i = b_i(s_i \oplus p_i) = b_i(s_i \oplus (\overline{a_i}\overline{b_i})) = b_i s_i$$

Applications

Task	Operator	Intermediates	Application
Addition	ADD	+	Arithmetic
Wide AND	AND	-	Word Equivalence (Match)
Wide OR	OR	-	Word Antivalence (Mismatch)
First One	OR	+	Arbitration, Arith. Normalization
Token Forwarding	ADD	+	Arbitration, Bit Masking
GRAY Decoding	XOR	+	Code Conversion
Saturated Bit Counting	OR^+	-	Voting

Use for general-purpose logic requires automation!

Sample Synthesis Results

Obtained for the example token propagation on the Xilinx Spartan-3 xc3s200-4 with the standard ISE workflow.

Impl.:	General-Purpose		Carry Chain			
Bit	Equations		Manual MUXCY		by Addition	
Width	LUTs	MHz	LUTs	MHz	LUTs	MHz
16	49	170.7	16	143.5	17	187.2
32	125	129.8	32	129.8	33	156.5
64	291	93.5	64	92.4	65	106.8
128	409	88.3	128	89.4	129	97.0

Note: Manual implementation can be revised to match implementation through addition.

Improvements in:
- Space (LUTs) *and*
- Time (MHz, Speed).

World Record

- First to complete the exploration of the 26-Queens Puzzle exploiting the exemplified token propagation scheme in a local FPGA setup. (**http://queens.inf.tu-dresden.de/**):
  → Discovered 22,317,699,616,364,044 solutions after 10 months of computation.
- The use of carry chains almost doubled the computation rate of the individual devices.
- Passed a world-wide grid computation, which is still up to about a year of computation.

References (excerpt)

[1] R. E. Ladner and M. J. Fischer, "Parallel prefix computation," *J. ACM*, vol. 27, no. 4, pp. 831–838, 1980.

[2] K. Vitoroulis and A. Al-Khalili, "Performance of parallel prefix adders implemented with FPGA technology," in *Circuits and Systems, 2007. NEWCAS 2007. IEEE Northeast Workshop on*, Aug. 2007, pp. 498–501.

[3] M. T. Frederick and A. K. Somani, "Beyond the arithmetic constraint: depth-optimal mapping of logic chains in LUT-based FPGAs," in *FPGA '08: Proceedings of the 16th international ACM/SIGDA symposium on Field programmable gate arrays*. New York, NY, USA: ACM, 2008, pp. 37–46.

[4] T. B. Preußer, B. Nägel, and R. G. Spallek, "Putting queens in carry chains," Fakultät Informatik, Technische Universität Dresden, Tech. Rep. TUD-FI09-03, Mar. 2009, ISSN 1430-211X, ftp://ftp.inf.tu-dresden.de/pub/berichte/tud09-03.pdf.