Efficient Multiple Regular Expression Matching on FPGAs based on Extended SHIFT-AND Method

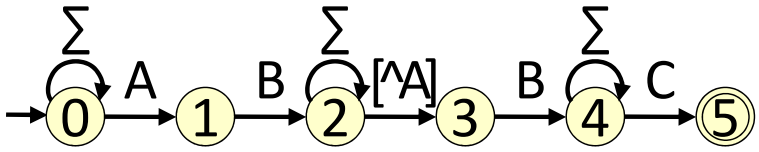
*Yusaku Kaneta, Shingo Yoshizawa, Shin-ichi Minato, [R4-8]
Hiroki Arimura, and Yoshikazu Miyanaga
(Graduate School of IST, Hokkaido University, Japan)

- **Large-scale pattern matching problem on hardwares**
  - A large number (e.g. thousands) of complex (e.g. regular expressions) against high-speed data streams (e.g. of several Gbps)
  - **Static compilation hardware** vs **Dynamic reconfigurable hardware**
- **Our Static BP-NFA architecture**
  - Static compilation hardware for a subclass of regular expressions, called linear regular expressions, based on bit-parallel method
  - Construction of our Static BP-NFA:

1. Linear regular expression: R

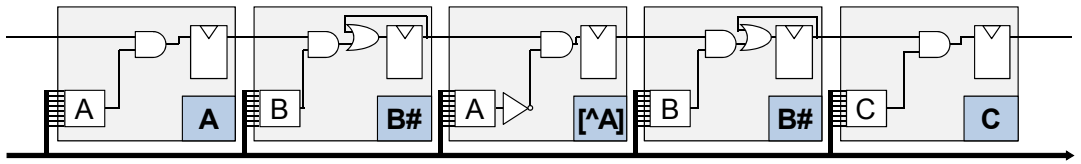
R = | A | B# | [^A] | B# | C |

2. NFA: N_R



3. Circuit on FPGA: M_R

Pattern matching module for R



Bit-parallel method is one of the pattern matching techniques on a general CPU

Experimental results

Performance evaluation and resource usage

- Throughput: **T = 1.6 Gbps**
- Number of PMMs: **N = 1500 PMMs** (19987 total chars)
- Compilation time: **C = 6000 sec** for **N = 1500**
  - Too expensive to frequently modify input patterns

Comparisons against other pattern matching hardwares

- Our Static BP-NFA for STR is faster than TNFA-based hardware and comparable with SHIFT-OR based hardware

Design	Class	Device	Throughput	Logic Cells/char	#Char Total
Static BP-NFA for STR (SASIMI2010, ours)	**STR**	Xilinx Virtex-5 LX50	1.6 Gbps	2.30	19887
TNFA-based hardware (FCCM2001)	**REG**	Xilinx Virtex 100	0.5 Gbps	66	29
SHIFT-OR based hardware (FPT2006)	**STR**	Altera Stratix EP1S40	2.3 Gbps	0.96	5004
Dynamic BP-NFA for STR (FPT2010, to appear)	**STR**	Xilinx Virtex-5 LX300	2.9 Gbps	10.8	8192
Dynamic BP-NFA for EXT (FPT2010, to appear)	**EXT**	Xilinx Virtex-5 LX300	1.6 Gbps	24.6	4096

STR: string patterns, **EXT**: extended string patterns, **REG**: regular expressions