

Hardware Acceleration of Operating Systems (Real-Time Task Scheduling)

Faster Task Scheduling

Lower Overhead of the Operating System

Higher CPU Utilization

More Real-Time Tasks Meet Their Deadlines

2 Custom Instructions

- `task_schedule` – inserts a new task into the queue ordered according to the deadline
- `task_kill` – removes any existing task from the queue according the ID of the task

Chip Area Costs

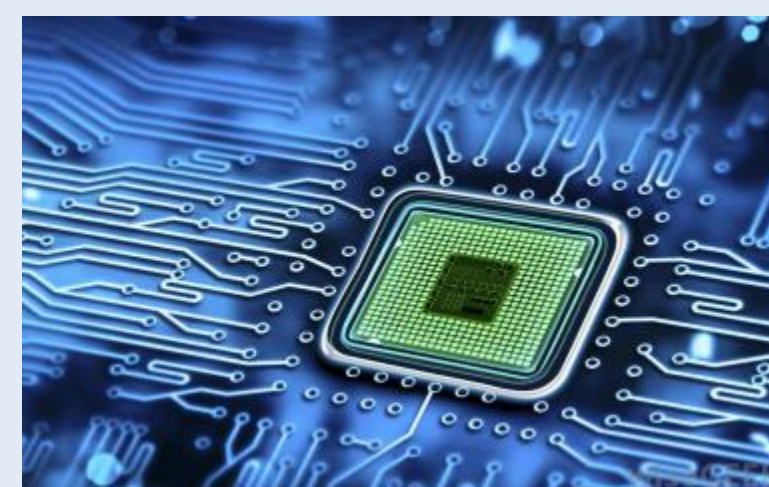
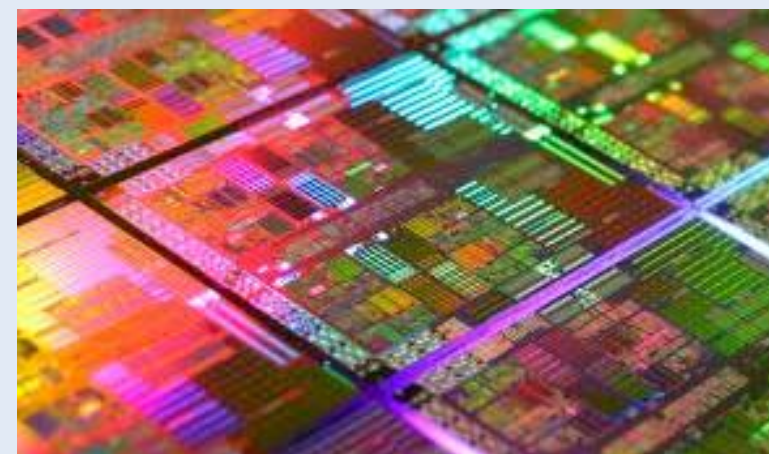
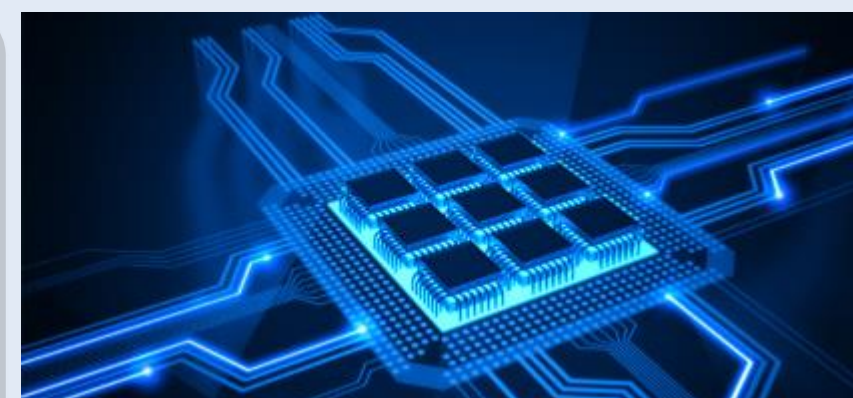
	control unit	per task	systolic array node	total*
D-FF	1	$\log(N) + D_width$	$2 \text{ per_task} + 2$	957
MUX	1	$3 (\log(N) + D_width)$	0	2401
NAND	1	0	0	1
NOR	0	1	0	32
AND	0	1	0	32
OR	$D_width + 1$	2	0	85
XOR	0	$\log(N)$	0	160
COMP	0	1	0	32
DEC	0	1	0	32

* - for $N = 32$, $D_width = 5$ and 4-stage pipelining

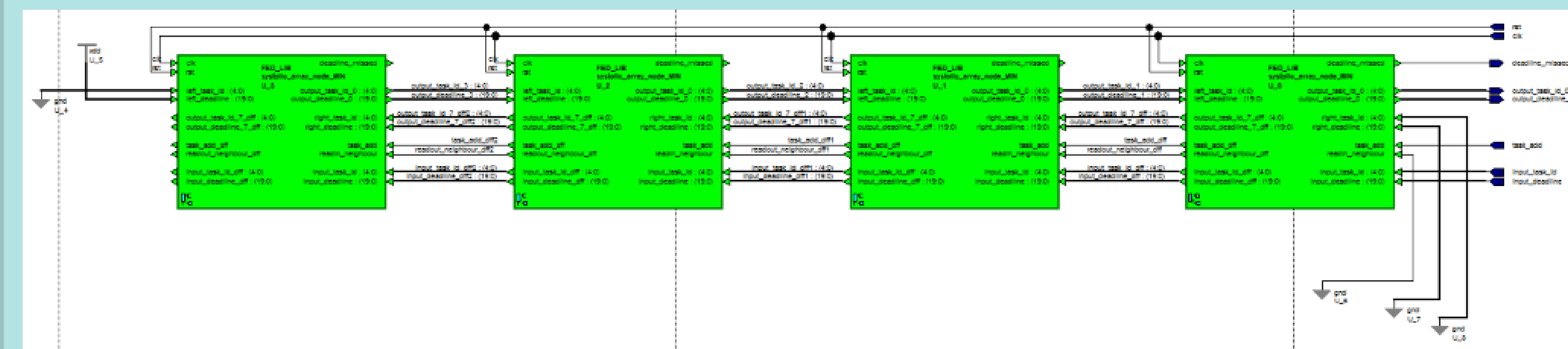
N – maximum number of tasks

D_width – bit width of the deadline times

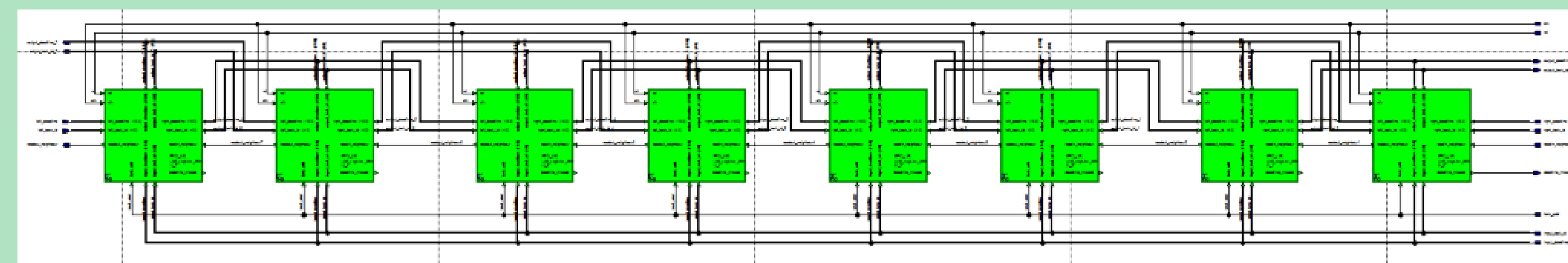
$\log(N)$ – bit width of the task ID



Serial Architecture – Pipelining



Parallel Architecture – Shift Registers



Speed Comparison

	software	hardware
asymptotic	$O(N)$	$O(1)$
clock cycles	$kN + c$	2

N – number of tasks in the system

FPGA Synthesis Results

Device Family: Altera Cyclone II

Device: EP2C70F896C6

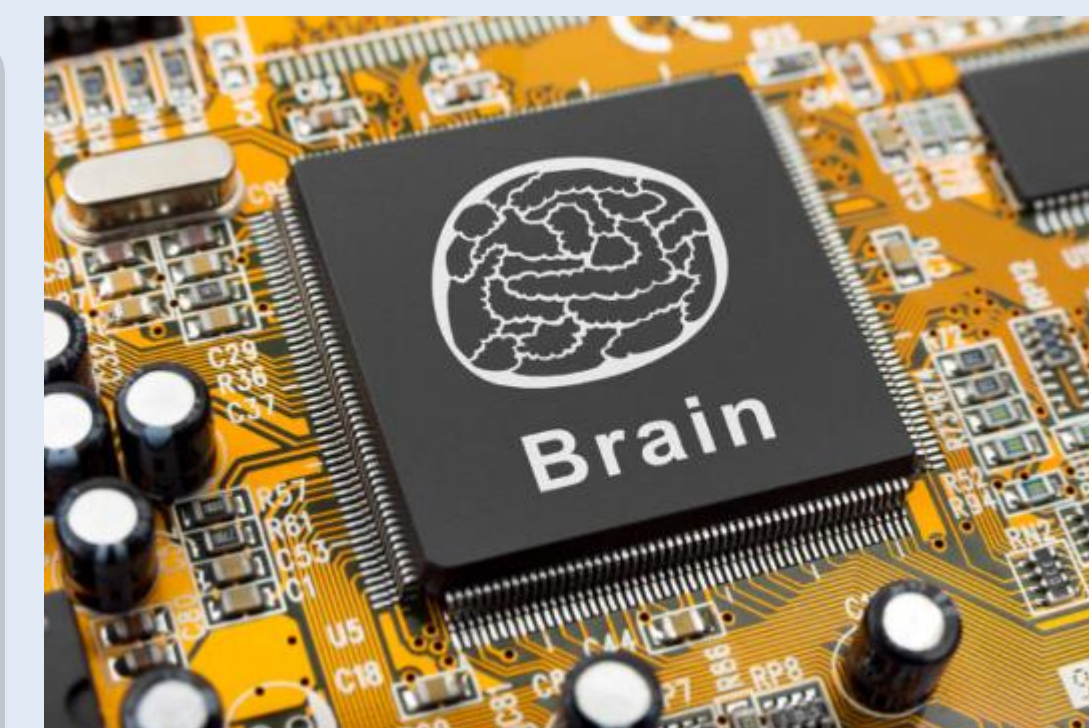
Logic Elements: 4547 (7%)

Combinational Functions: 3912 (6%)

Dedicated Logic Registers: 957 (1%)

Max. Number of Tasks: 32

Deadline Accuracy: 20 bits (1 us)



Bc. Lukáš Kohútka

Verification

- Pre-Layout Simulations
- Software Implemented Simulations (C Language Console Application)
- Functional BIST - synthesizable verification