



Implementing TTA ASIPs on Standard Cell Technology

Jaakko Sertamo

jaakko.sertamo@tut.fi

Institute of Digital and Computer Systems

Tampere University of Technology

Tampere, Finland



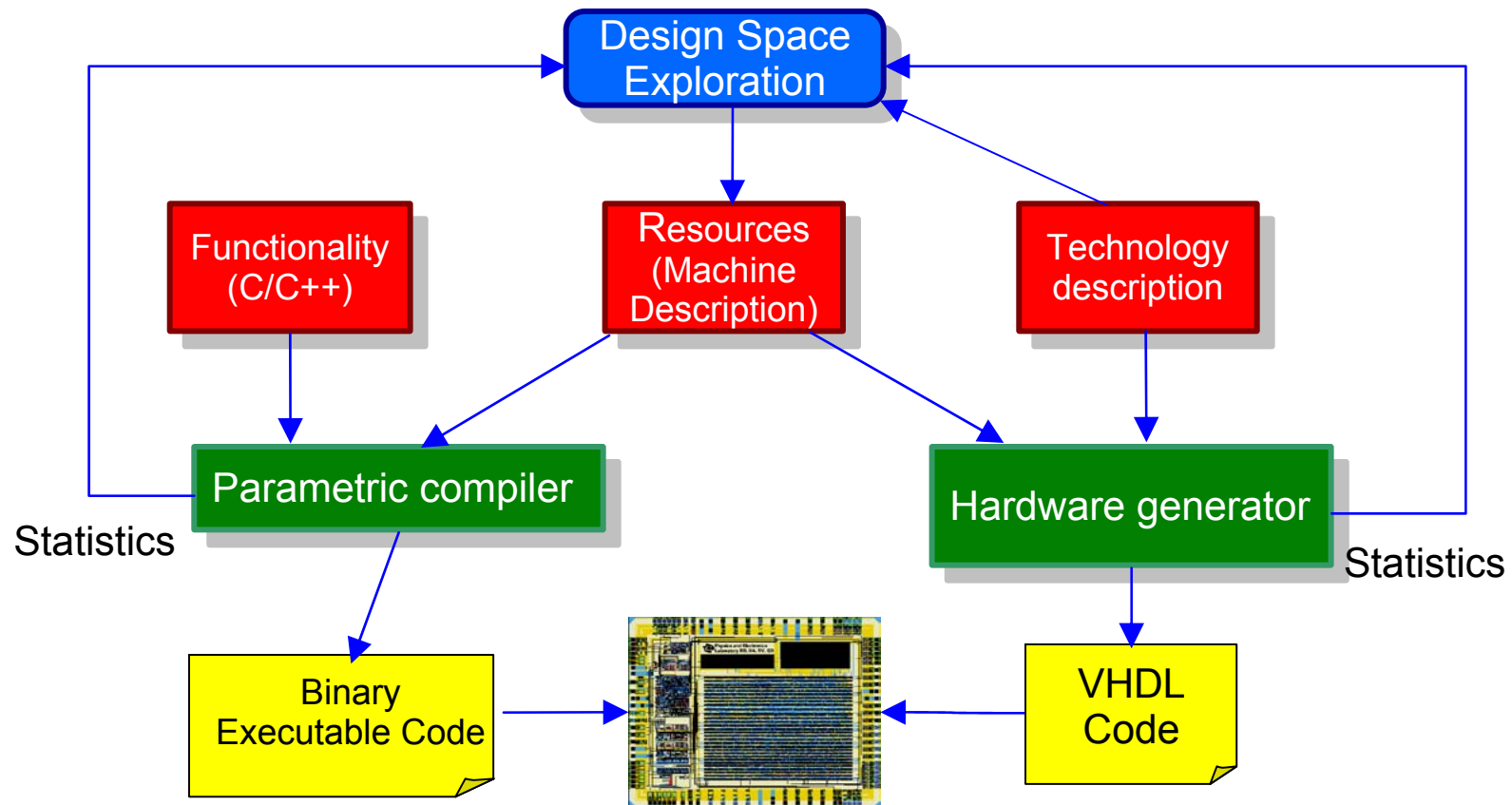
Outline

- TTA ASIP design flow @ TUT/DCS
 - Optimization methods for TTA ASIP implementations on standard cells
 - Replacement strategies for the three-state bus
 - Clock gating
 - Examples
-



TTA ASIP design flow @ TUT/DCS

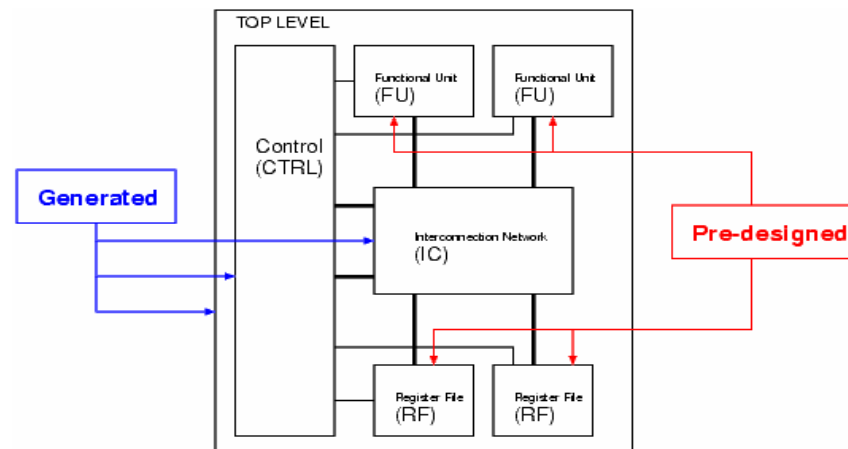
- Based on MOVE Framework IDE developed at TUD





TTA ASIP design flow @ TUT/DCS

- After a suitable architecture configuration is found, VHDL description of the target processor core is generated using processor generator scripts
 - Scripts, written in Python, are not part of the MOVE framework
 - They perform HDL to HDL transformation from internal machine description format to synthesizable VHDL code
 - Leaf cells, i.e, functional units and register files are not generated
 - Implementation methodology of leaf cells can be freely chosen





TTA ASIP design flow @ TUT/DCS

- Standard cell implementation approach
 - Because of the simple control required in the functional units (SVTL pipelining) a full range of units supporting majority of integer operations can be easily described in RTL VHDL
 - VHDL code is useful in two ways
 - Simulation/Verification at RTL level
 - Input for the logic synthesis
 - Also simple scripts for the logic synthesis can be obtained from the processor generator
 - Synthesis itself is a fairly straightforward procedure
 - Simple top-down strategy is used
 - However, we are still far from working silicon
 - Little experience from complex, tool/vendor dependent physical design flow
-



Optimization methods for TTA ASIP implementations on standard cells

- ❑ Modularity of TTAs makes HDL writing / generation straightforward
 - No redundant logic or registers in generated HDL code
 - ❑ Most of the optimizations are performed automatically by the synthesis tool
 - Tool used: Synopsys Design Compiler
 - Automatic selection of the accurate architecture of arithmetic units
 - For example, CSA vs. Wallace-tree multiplier
 - In general, describing structural logic (arithmetic units) independent of synthesis tool is very difficult and gives small benefits
-



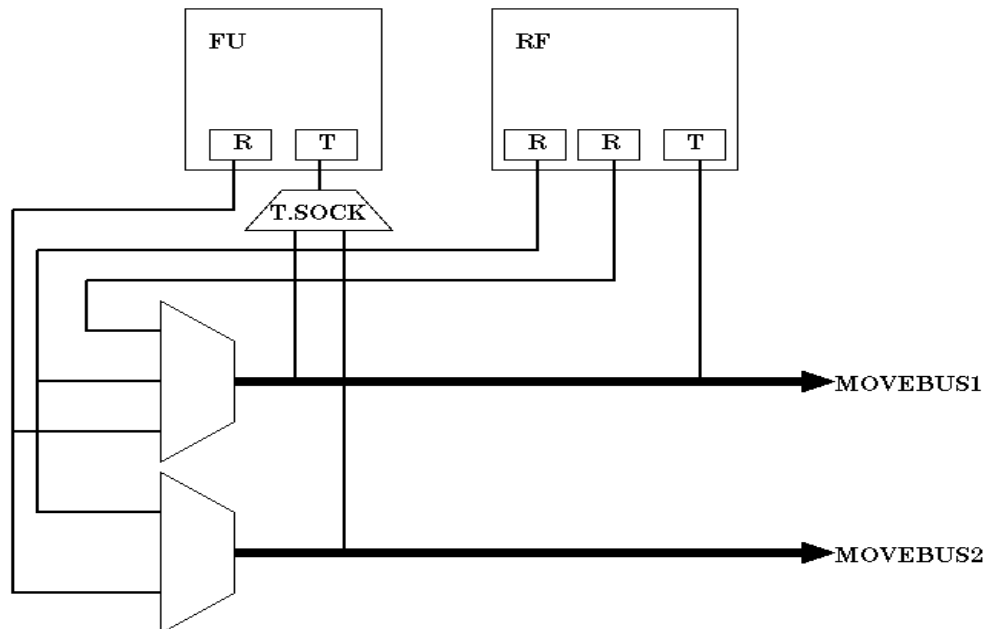
Optimization methods for TTA ASIP implementations on standard cells

- ❑ Traditionally, transport (interconnection) network has been proposed to be implemented using three-state buffers
 - May result in smaller wiring area in full custom designs when only few metal layers are available
 - Nowadays every process has 5+ layers of metal
 - Place and Route may give better results when no artificial boundaries between units and busses are given
 - Possible reliability problems
 - ❑ Fortunately, there are alternative topologies to replace the three-state bus
-



Optimization methods for TTA ASIP implementations on standard cells

- ❑ Multiplexer based selection for bus writes
 - Each FU or RF output connected to a bus is an input to a multiplexer that drives the bus
 - No result sockets
 - Several alternatives for implementing the multiplexer
 - Does not affect the programming model

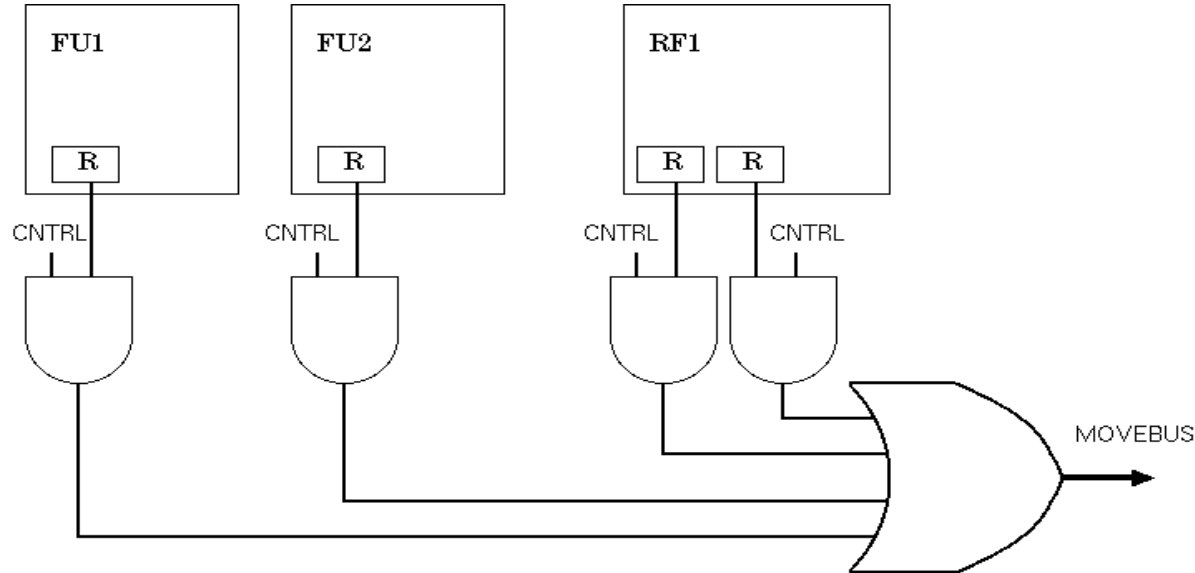




Optimization methods for TTA ASIP implementations on standard cells

□ AND-OR structure

- Control bit for each output connected to a bus
- Described explicitly in HDL
- The final selection of the used cells is performed by the synthesis tool





Optimization methods for TTA ASIP implementations on standard cells

- ❑ Traditional three-state drivers
 - ❑ “HDL multiplexer”
 - Source field from instruction word used to control which output writes to the bus
 - Synthesis tools selects the the detailed implementation
 - Results in more or less same structure as in AND-OR based bus
 - Selection probably suffers from combined ID and opcode in source field
 - More bit patterns in SRC field, that than inputs to selection logic
 - They cannot be considered as 'don't cares'
-



Optimization methods for TTA ASIP implementations on standard cells

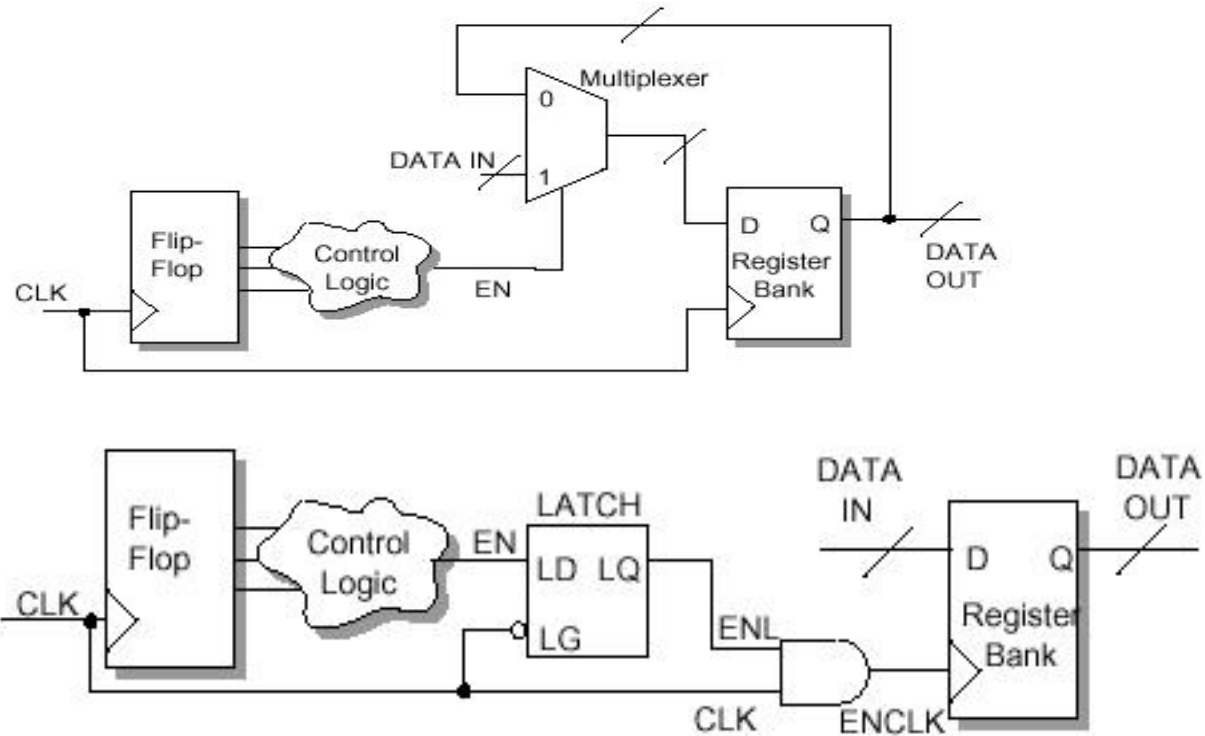
Clock gating

- Goal: decrease the amount of nodes to be charged/discharged at clock transition
 - Can be performed automatically by the synthesis tool
 - HDL coding style may improve results
 - Basic Idea: Make as many register writes as possible conditional
 - Modularity of TTA promotes clock gating
 - No clock activity in units that are not used
 - Drawbacks
 - Makes clock tree synthesis (CTS) more complex
 - Synthesis tool dependent
-



Optimization methods for TTA ASIP implementations on standard cells

- Reduction in **both** power and area



Reference: Synopsys, Power Compiler Reference Manual



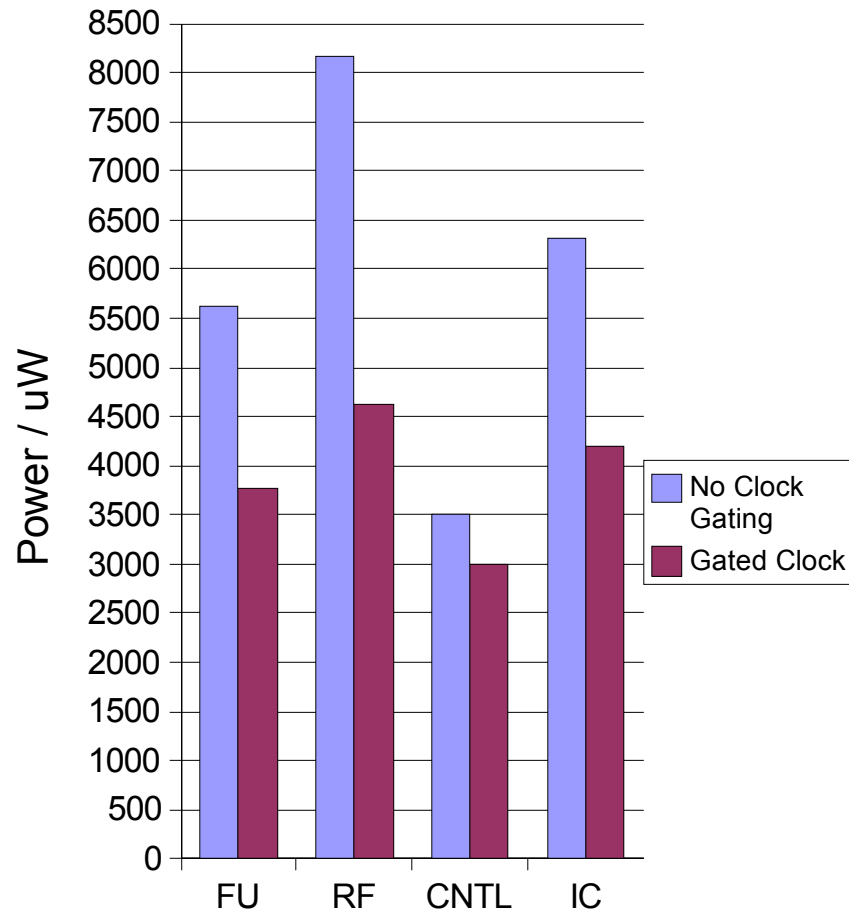
Examples

- ASIP for 32-point DCT (discrete cosine transform) application
 - Comparison: No clock gating vs. Gated clock
 - Architecture configuration
 - 9 busses
 - 3 adders, shifter, multiplier, 2 load-store units
 - 8 X 4 registers
 - Simulation parameters
 - Clock frequency 100 MHz
 - Operation voltage 1.5 V
 - Nominal operating conditions ($T=25^{\circ}\text{C}$)
-



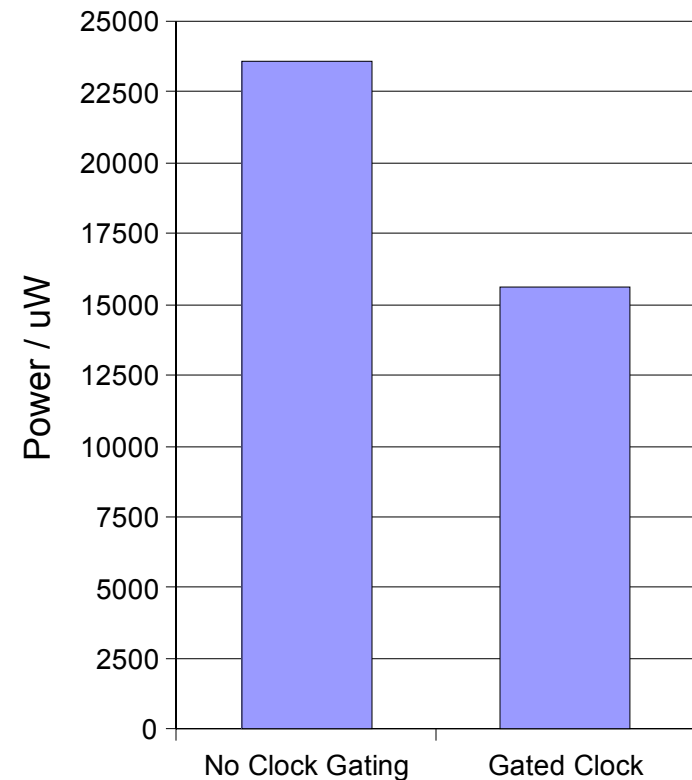
Examples

Power consumption by components



Total Power consumption

○ Saved 33.9 %



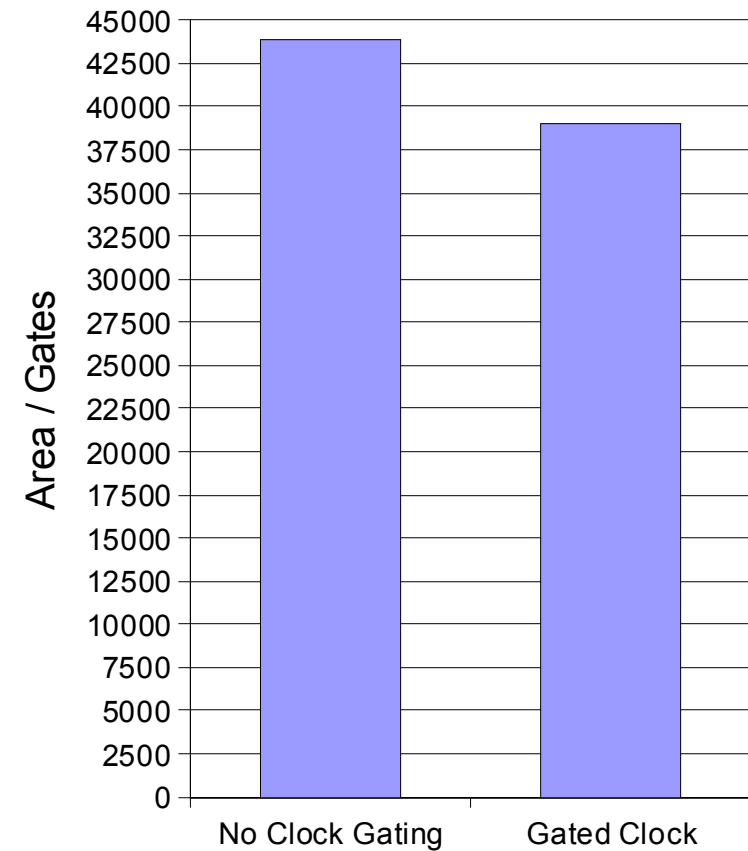
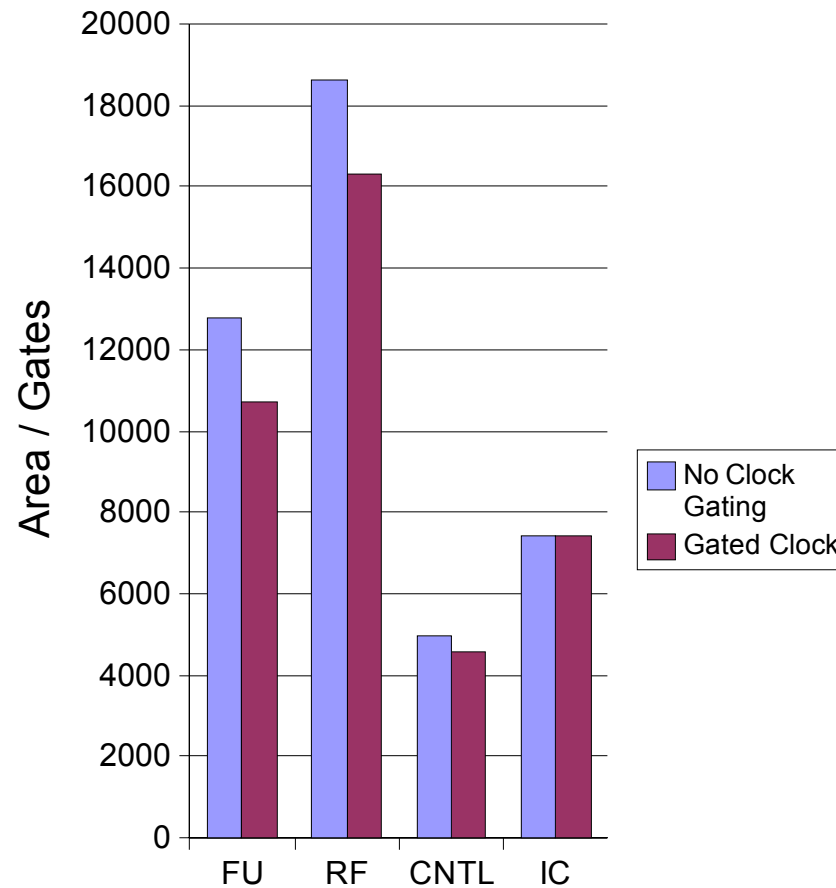


Examples

□ Area by components

□ Total area

○ Saved 10.9 %





Examples

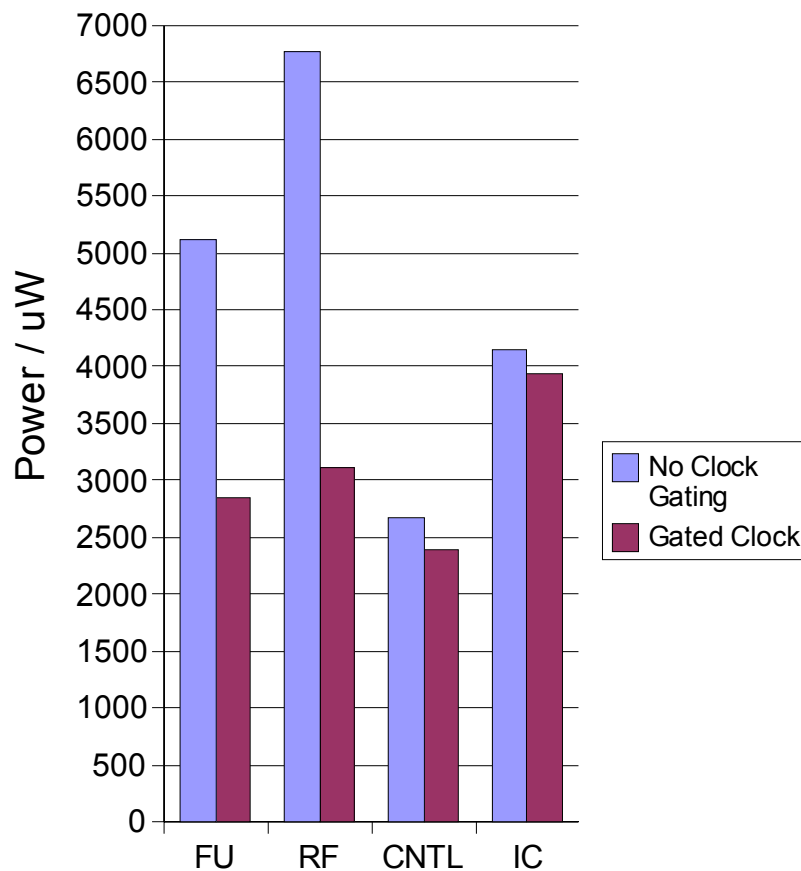
□ ASIP for 8x8 2D-DCT

- Comparison: No clock gating vs. Gated clock
 - Architecture configuration
 - 7 busses
 - 2 adders, shifter, multiplier, comparator, 2 load-store units
 - 8 X 4 registers
 - Simulation parameters
 - Clock frequency 100 MHz
 - Operation voltage 1.5 V
 - Nominal operating conditions ($T=25^{\circ}\text{C}$)
-



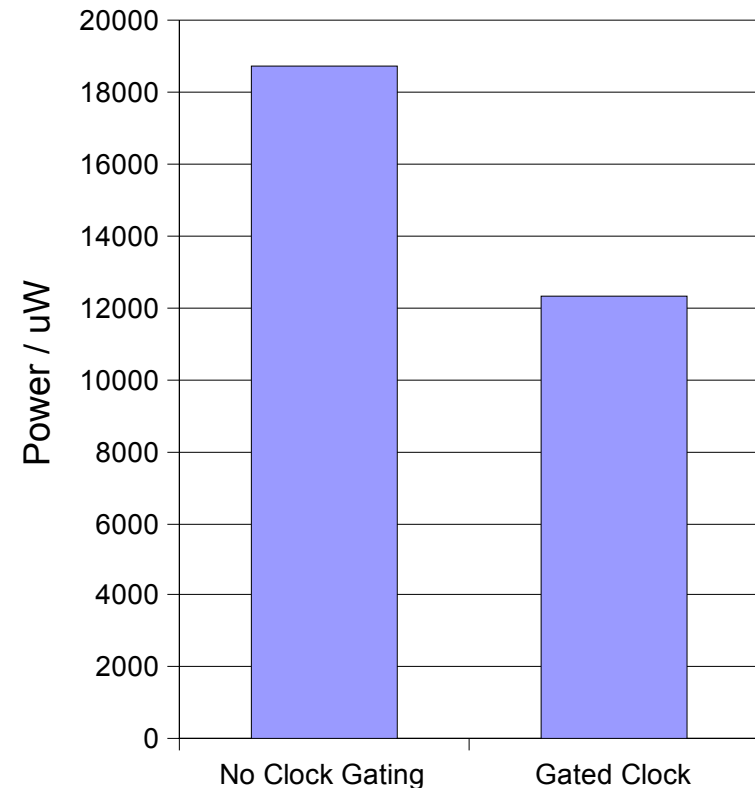
Examples

❑ Power consumption by components



❑ Total power consumption

○ Power saved 34.2%

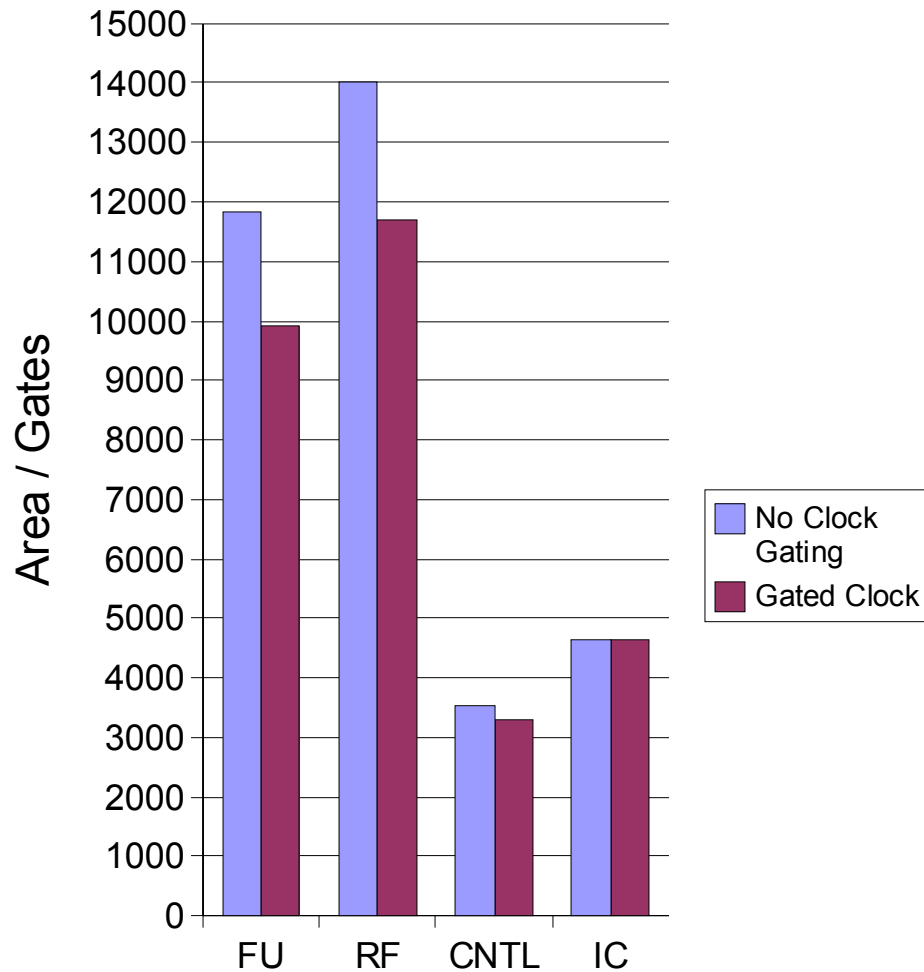




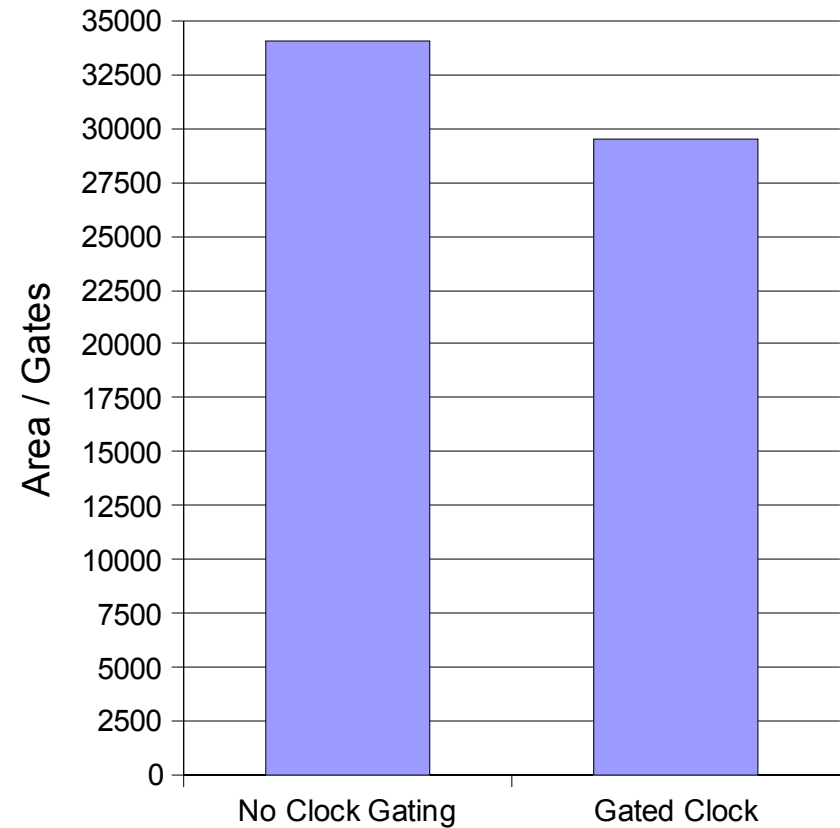
Examples

□ Area by components

□ Total area



○ Area saved: 13.2%





Examples

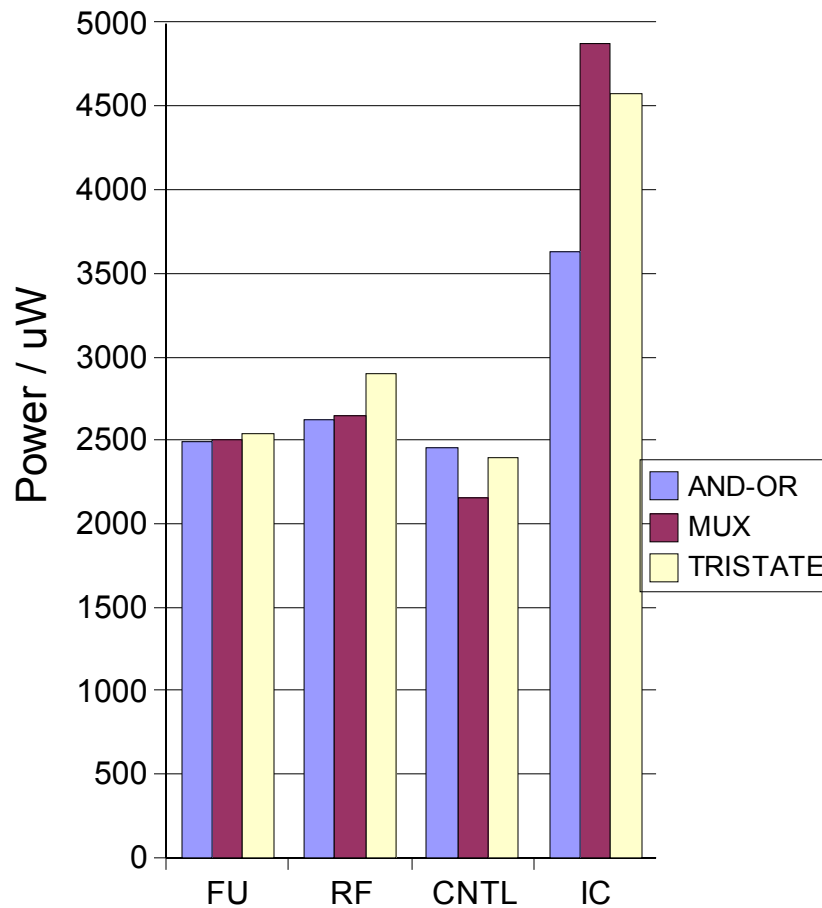
□ ASIP for 8x8 DCT

- Comparison: multiplexer/bus write implementations
 - Architecture configuration
 - 7 busses
 - 2 adders, shifter, multiplier, comparator, 2 load-store units
 - 8 X 4 registers
 - Simulation parameters
 - Clock frequency 100 MHz
 - Operation voltage 1.5 V
 - Nominal operating conditions ($T=25^{\circ}\text{C}$)
-

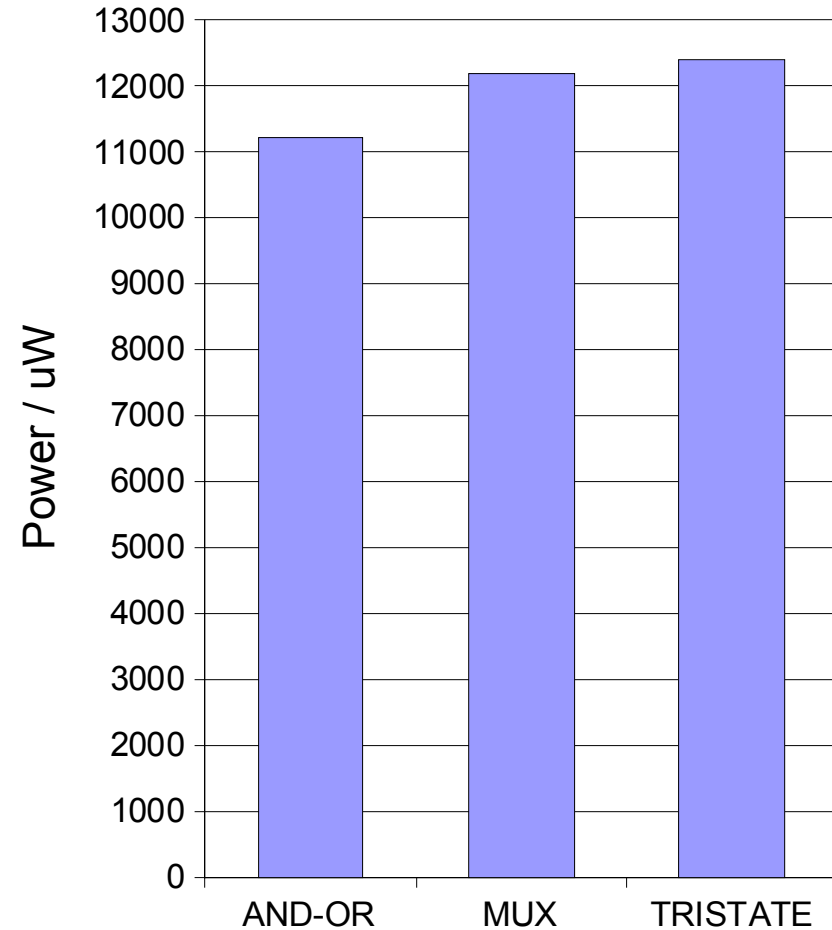


Examples

Power consumption by components



Total Power

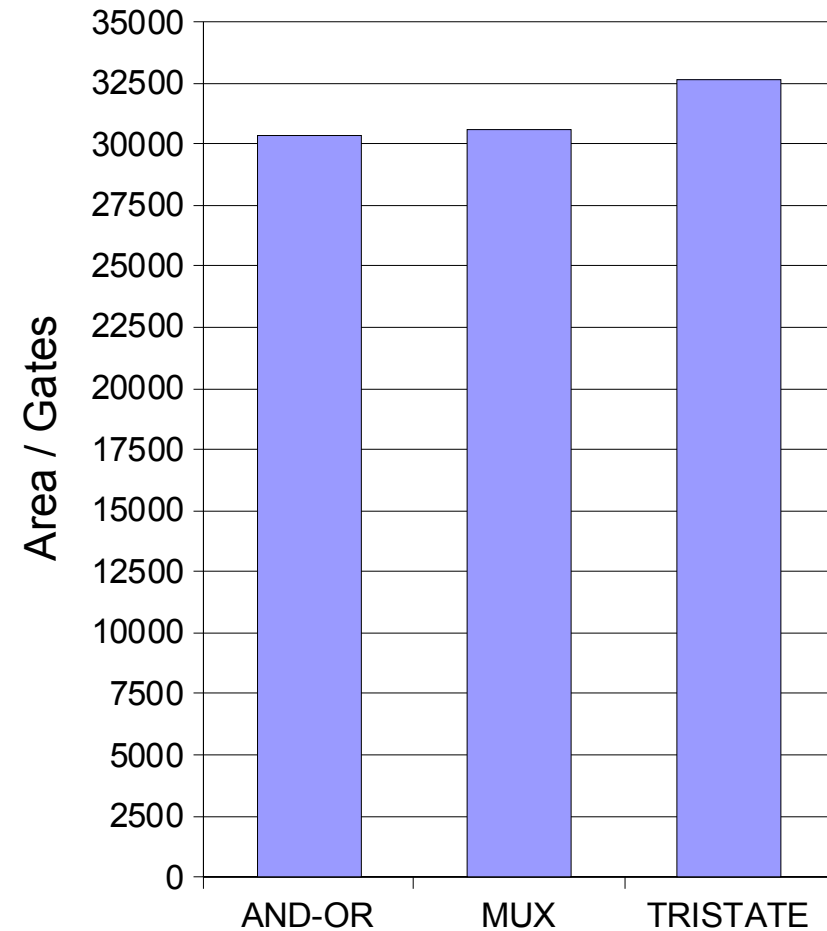
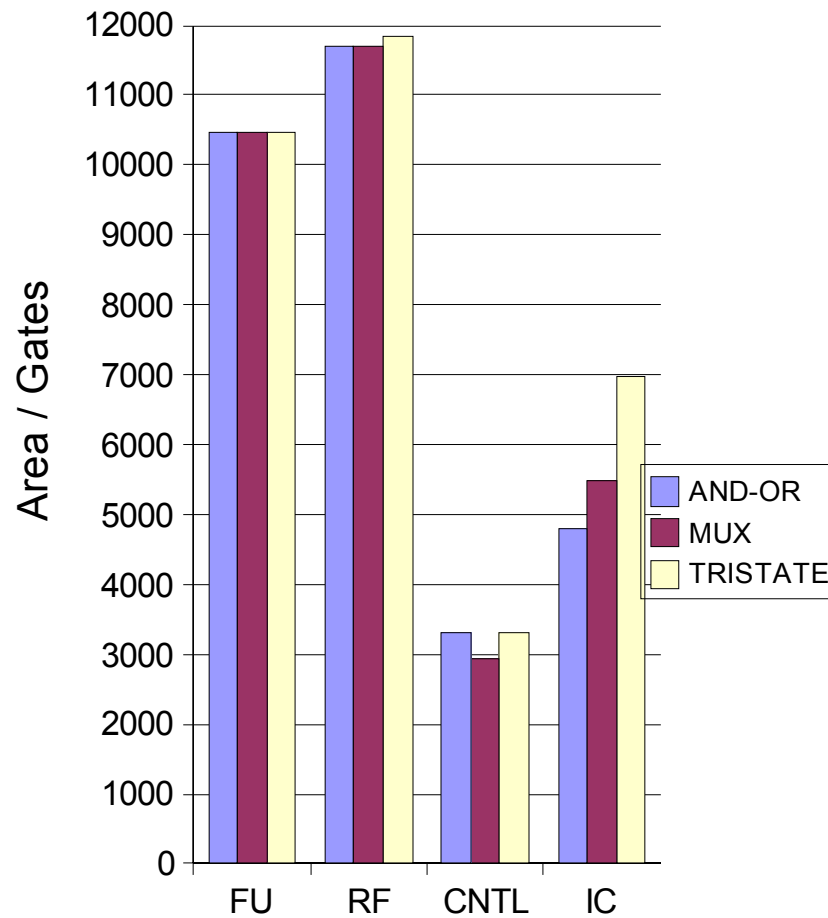




Examples

□ Area by components

□ Total Area





Conclusions

- Most of the design optimizations are performed at application/architectural level
 - Register files are the most critical part of the TTAs implemented with standard cell technology
 - There are alternatives for three-state bus which result in smaller area and lower power consumption
 - Clock gating can significantly reduce power consumption of TTA ASIPs
-