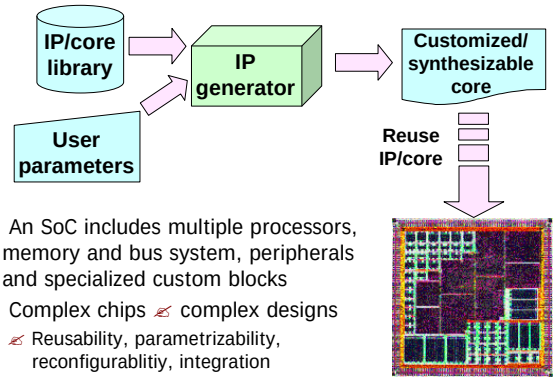
5P-11

PARLAK: Parametrized Lock Cache Generator

Bilge E. S. Akgul and Vincent J. Mooney III
School of Electrical and Computer Engineering
Georgia Institute of Technology, USA

Introduction

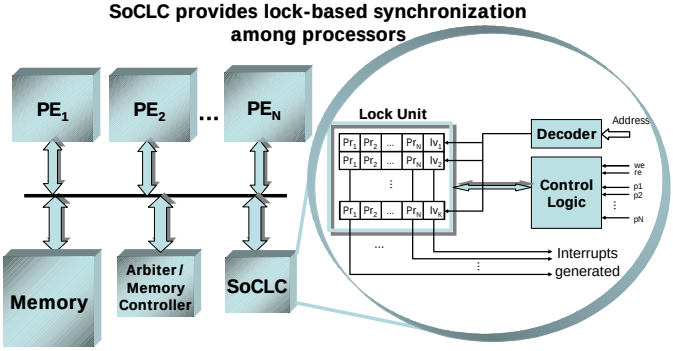
Intellectual Property (IP) or core based design for System-on-a-Chip (SoC)



- An SoC includes multiple processors, memory and bus system, peripherals and specialized custom blocks
- Complex chips ✍ complex designs
  - Reusability, parametrizability, reconfigurablitiy, integration
  - Engineering effort
  - Time-to-market
- Solution: IP-generator tools
  - Automated generation of synthesizable IP blocks and system components

Custom hardware: SoC Lock Cache (SoCLC)

SoCLC provides lock-based synchronization among processors

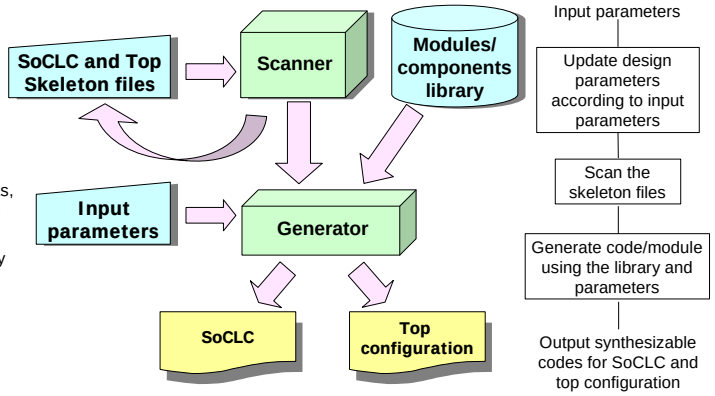


- SoCLC is an IP core that provides effective lock-based synchronization for a shared-memory multiprocessor SoC
- Lock variables are accessed from the SoCLC
  - Two types of locks: Short critical section locks and long critical section locks
  - Interrupts are generated to notify processors when a lock is released
  - Achieved speedups of 27% to 55% in practical examples
  - More information at http://codesign.ece.gatech.edu/publications/

Methodology

- PARLAK is an IP-generator tool for SoCLC
  - Generates customized, user specified versions of SoCLC
  - Useful for after/prior to chip fabrication
- Building blocks:
  - Input parameters
    - Number of short critical section locks, number of long critical section locks
    - Number and types of processors
    - Interrupt generation (FIFO or priority based)
  - Skeleton files (in Verilog HDL)
    - Signal, process and module descriptions (independent from the input parameters)
  - Library of modules/components

PARLAK: Parametrized Lock Cache Generator



Sample skeleton input file and generated output file

Skeleton file

```
Module LockCache (clk,rst,re,we, ... , D,A,IRQ,pr);
//LABEL #1 – PARAMETER declarations
...
//LABEL #2 – Interrupt lines updated with priority
...
//LABEL#3 – Control logic instantiations
...
```

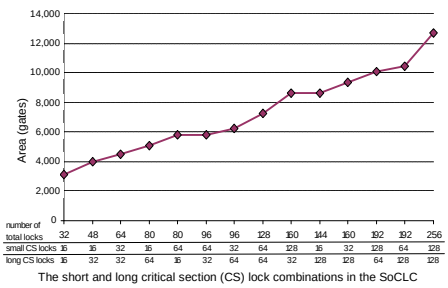
Output file

```
Module LockCache (clk,rst,re,we, ... , D,A,IRQ,pr);
//LABEL #1 – PARAMETER declarations
parameter NUM_PE = 2;
parameter SMALL_LOCKS = 64;
parameter LONG_LOCKS = 32;
parameter ADDR_W = 5;
//LABEL #2 – Interrupt lines updated with priority
IRQ[0] = irq_q[0];
IRQ[1] = irq_w[1] & ~irq_w[0];
//LABEL#3 – Control logic instantiations
control ctrl1 (.clk(clk), .rst(rst), .in(inctrl1),
.out(irq_w[0]), .we(we));
control ctrl2 (.clk(clk), .rst(rst), .in(inctrl2),
.out(irq_w[1]), .we(we));
```

Synthesis Results of SoCLC Configurations Generated Using PARLAK

- Full range of customized SoCLCs that are generated by PARLAK have been directly synthesized using Design Compiler from Synopsys
  - An SoCLC for two processors with 32 lock variables occupies 1,790 gates and an SoCLC for 14 processors with 256 lock variables occupies 37,380 gates
  - 0.25? technology TSMC standard cell library from LEDA is used
- PARLAK output SoCLC and top configurations are also simulated to test correctness in the Seamless CVE platform from Mentor Graphics
  - Four MPC750 processors, shared memory, SoCLC, decoder and arbiter

Synthesis results of SoCLC for increasing number of locks. Number of processors = 4.



Synthesis results of SoCLC for increasing number of processors that the SoCLC is configured for. Number of lock variables is changed from 32 to 256.