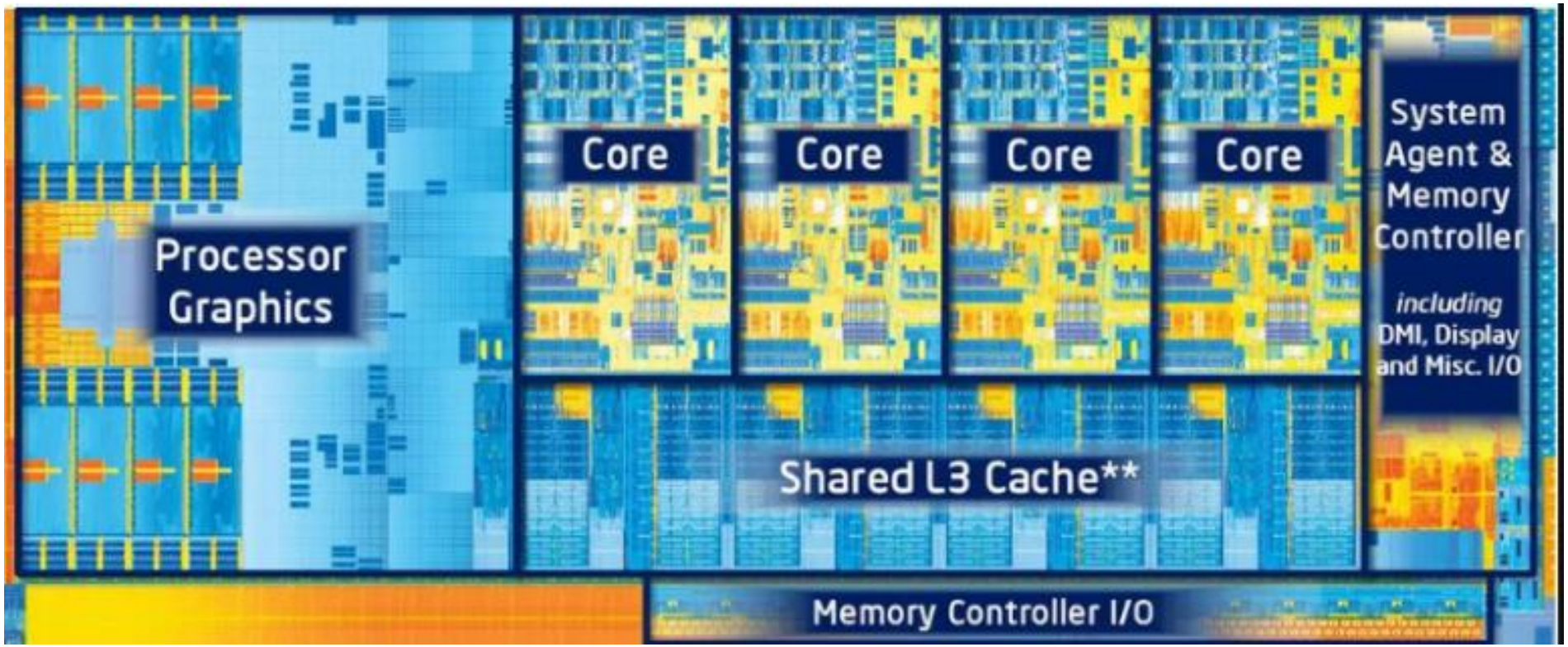
Lab on ADVANCED VLSI DESIGN

Introduction

2025

A Class on Optimizing the Critical Piece

- **Why do we work on Full Custom Design in Digital Circuits?**



Mastering Ful-Custom Digital

5.7 A 9GHz 65nm Intel Pentium® 4 Processor Integer Execution Core

Sapumal Wijeratne, Nanda Siddaiah, Sanu Mathew, Mark Anders, Ram Krishnamurthy, Jeremy Anderson, Seung Hwang, Matthew Ernest, Mark Nardin

Intel, Hillsboro, OR

The 64b integer core of the Intel Pentium® 4 processor shown in Fig. 5.7.1 uses a 2× frequency clock to enable single-cycle latency on the critical ALU bypass loops [1]. Domino circuitry optimized for 2× frequency operation replaces low voltage swing (LVS) circuit technology [2], because process-scaling studies indicated that transitioning LVS to a 65nm CMOS process [3] will only yield a speedup of 14%–an amount that is insufficient to enable the frequency targets of a 4th-generation Intel Pentium® 4. The circuit architectures implemented in the ALUs, AGU, and integer register file enable a 9GHz/1.3V frequency target while reducing power consumption, relative to a 90nm version of this product described in [4].

segment base from the linear address, the extra carry-out is obtained 3 gate delays after the linear address. This technique minimally affects the critical early-arriving lower 16b address, needed for the cache tag, by loading only the non-critical 4b conditional sum circuits. The sparse-tree adder organization allows the subtract operation to begin in the conditional sum blocks (Fig. 5.7.4c) before the linear address computation is complete. Within each 4b conditional sum circuit, 2 additional ripple-carry chains, one with carry-in of 1 and the other with carry-in of 0, create conditional PG signals, respectively, for the subtraction. An additional MUX chooses the correct PG signals, which are then merged in 2 stages of CM shown in Fig. 5.7.4b to yield the MUX select for the upper 16b linear address. Compared to the LVS AGU design, the sparse-tree design, ripple-carry subtract, and efficient clock-gating enable 59% normalized dynamic power reduction and 10% area reduction.

A conventional rotator architecture using a single, wide MUX cannot perform single-cycle rotate/shift operations and meet bandwidth requirements at 2× frequency. As a result, the rotate/shift function is broken into 3 domino MUX stages (Fig. 5.7.5). The first MUX stage rotates each byte 0-7 positions, and the second MUX rotates 0, 8, 16

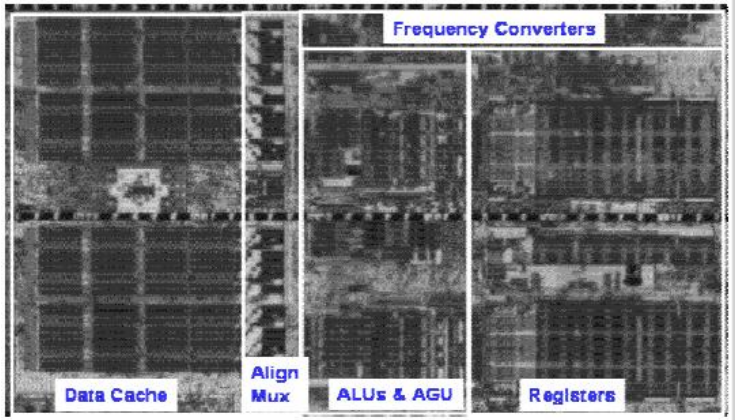


Figure 5.7.8: Die micrograph showing outlines of integer core blocks.

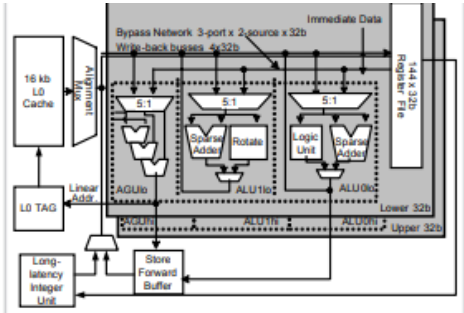


Figure 5.7.1: Integer execution core organization.

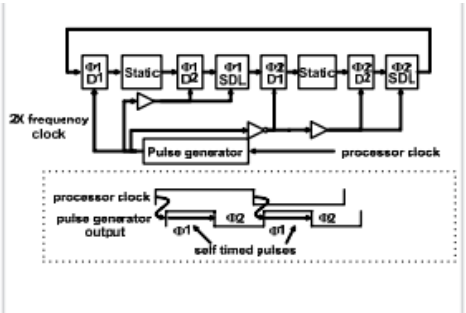


Figure 5.7.2: 2x frequency domino and clock scheme.

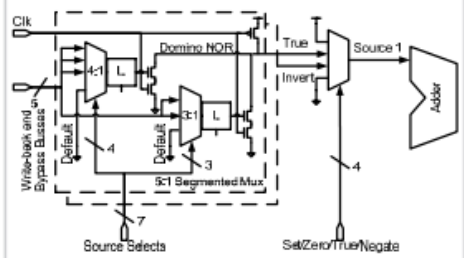


Figure 5.7.3: AGU/ALU front-end design.

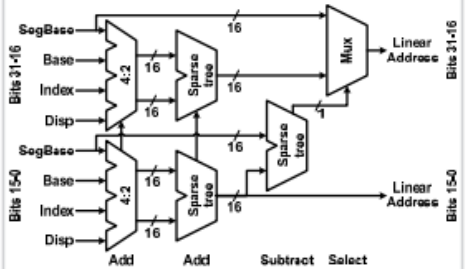


Figure 5.7.4a: AGU organization.

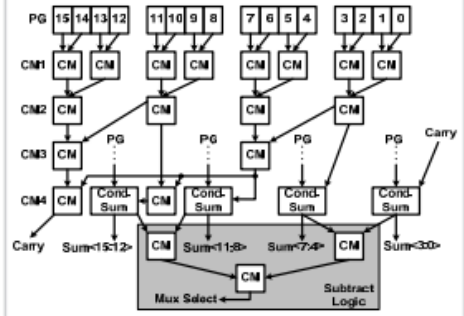


Figure 5.7.4b: Critical AGU 16b sparse carry-merge tree.

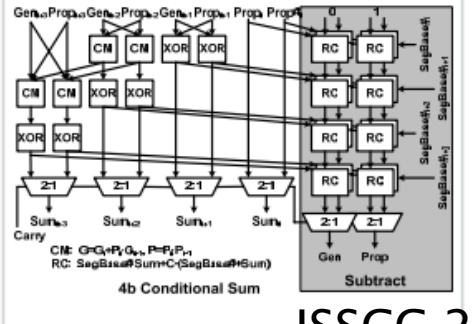


Figure 5.7.4c: AGU 4b conditional sum-generators.

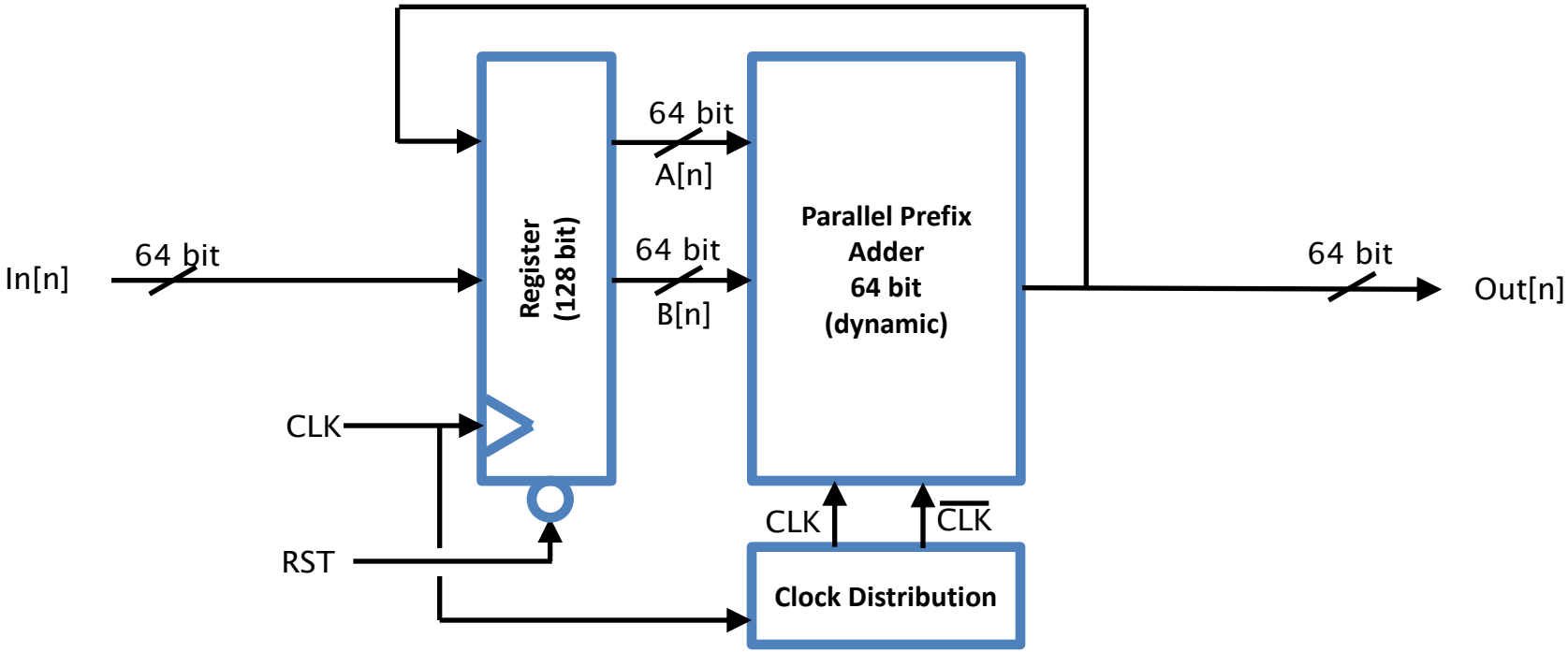
ISSCC 2006

Objectives of the Class

- **Collect hands-on experience with Full-Custom digital design**
- **Plan and optimize a challenging high-speed logic block on schematic level**
- **How to size transistors and identify speed bottlenecks**
- **Plan critical structures such as clock trees and routing**
- **Come up with a good floor plan**
- **Anticipate parasitics to optimize your schematic**
- **Draw and optimize a full-custom layout**

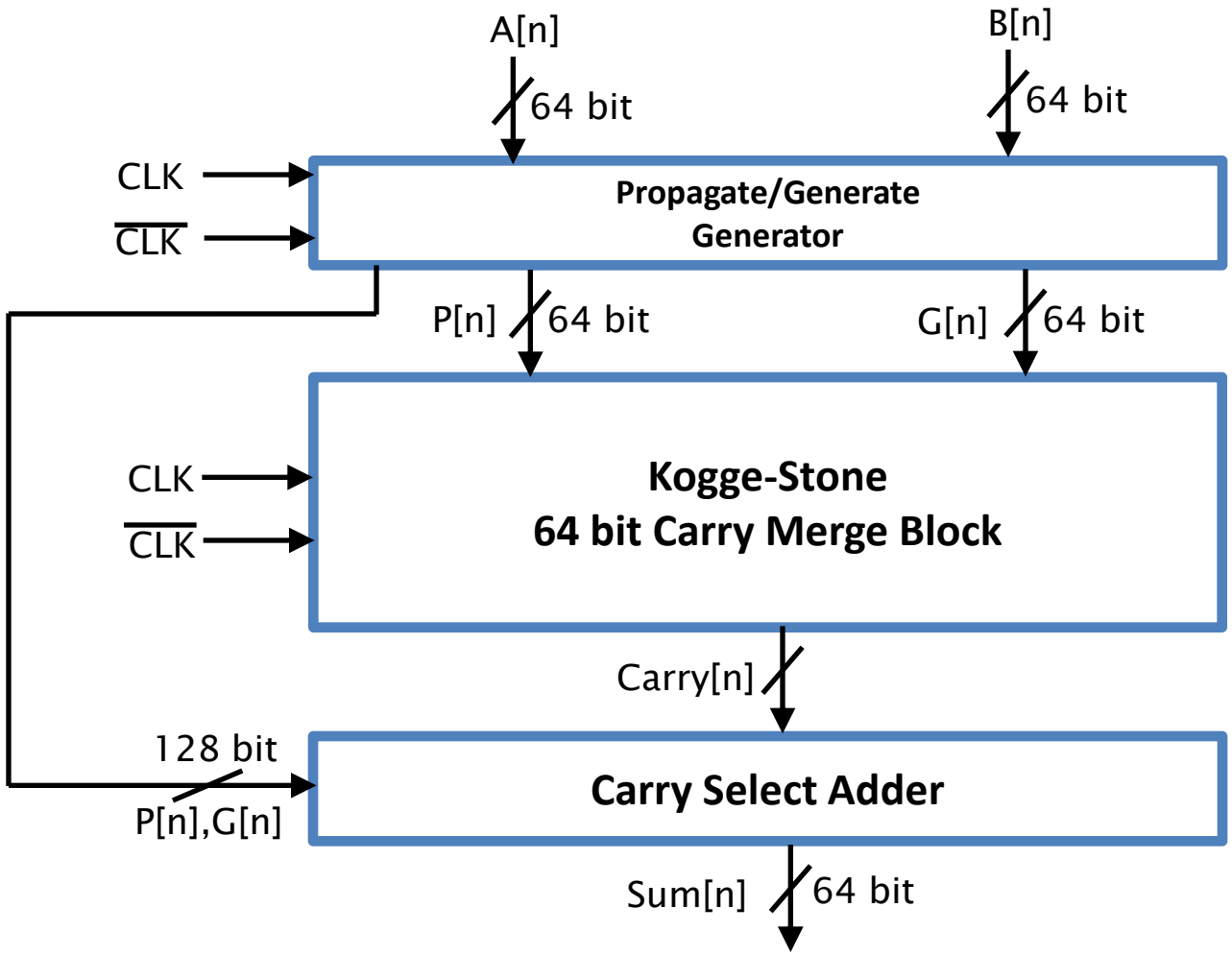
Project

64 Bit Parallel Prefix Adder with Dynamic Logic



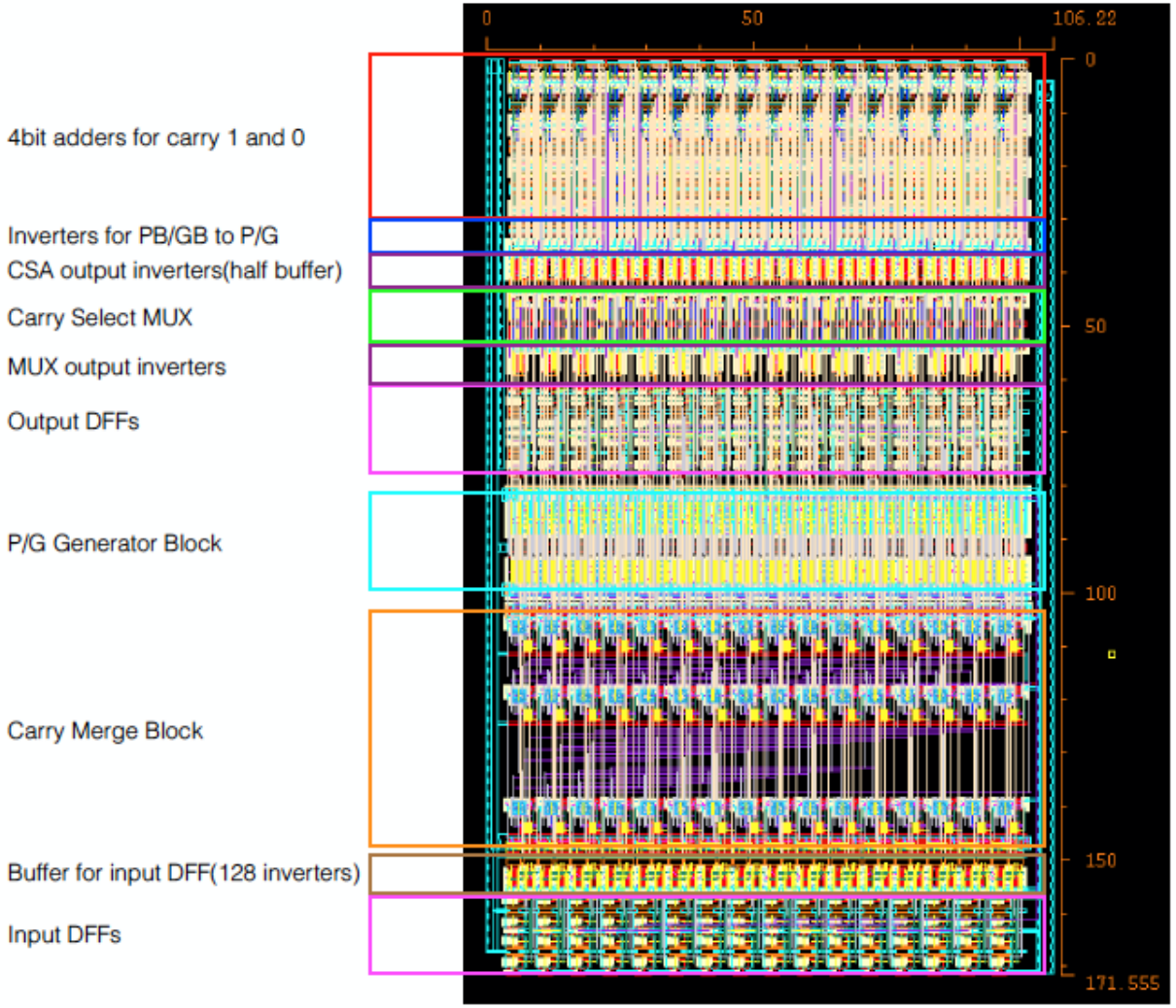
B[n+1] = In[n]
A[n+1] = Out[n]
Out[n] = A[n] + B[n]

Project



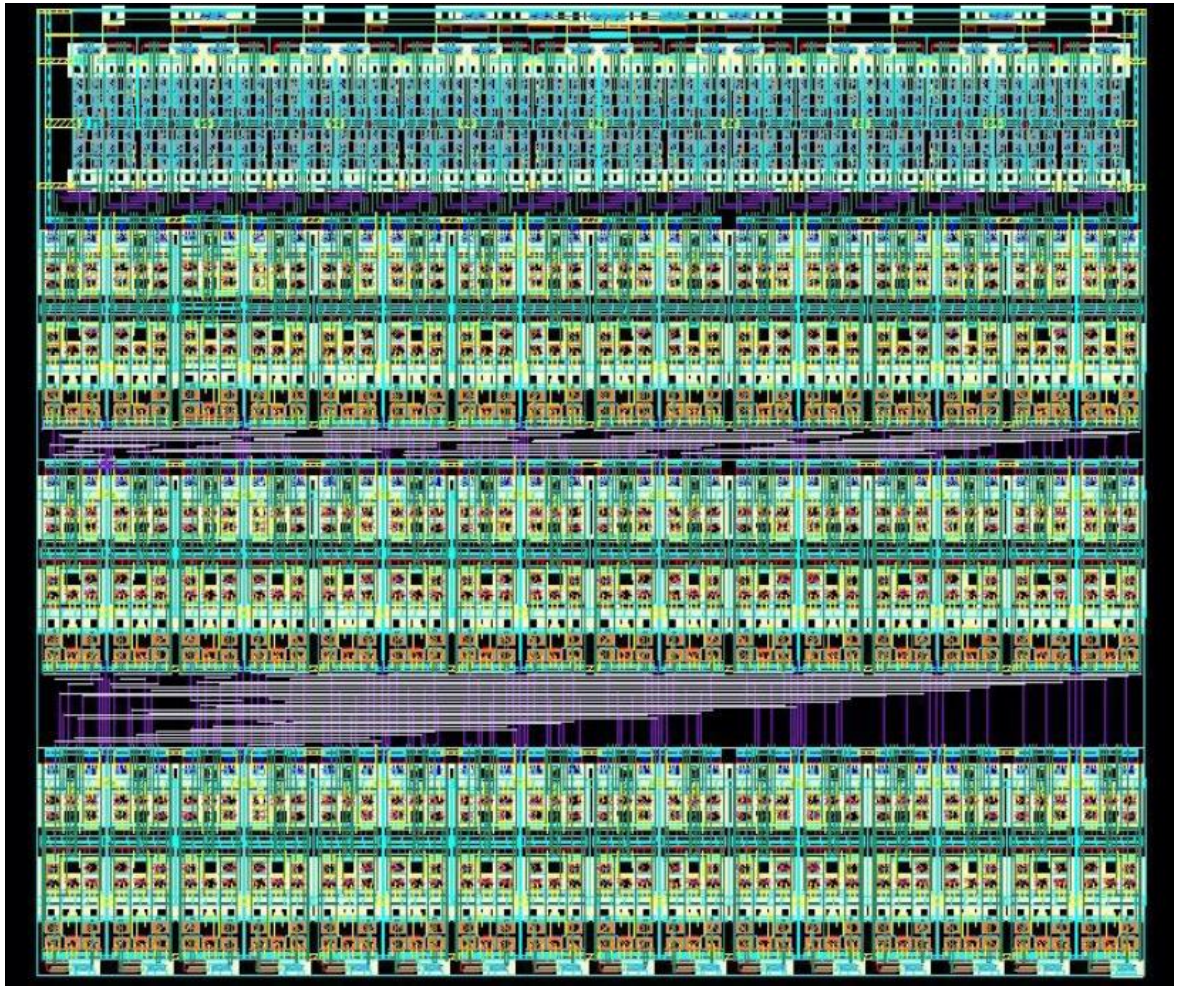
More details will be explained next week…

64-bit parallel-prefix Kogge-Stone adder/accumulator with dynamic (Domino) logic in 90nm CMOS technology



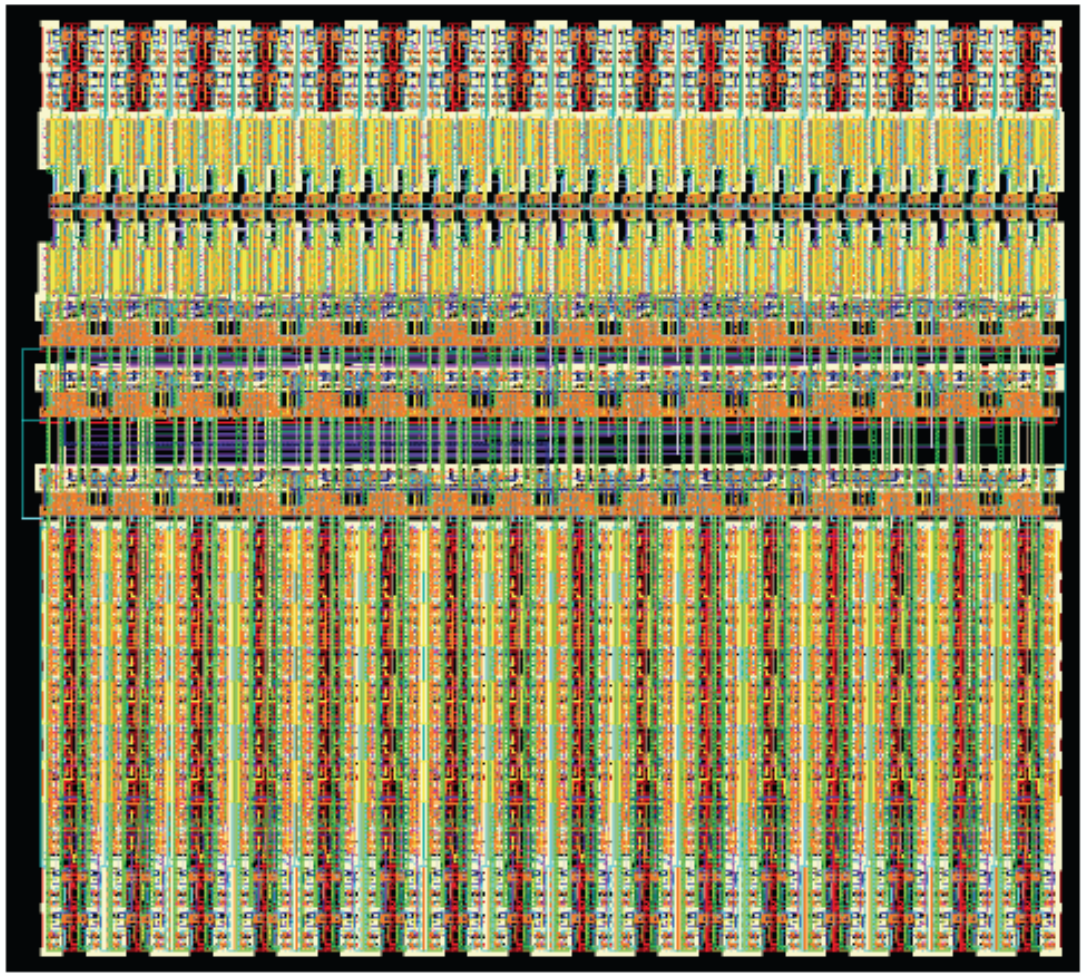
Design: Gain Kim

64-bit parallel-prefix Kogge-Stone adder/accumulator with dynamic (Domino) logic in 90nm CMOS technology



Design: Lucille van Som

64-bit parallel-prefix Kogge-Stone adder/accumulator with dynamic (Domino) logic in 90nm CMOS technology



Design: Lorentz Schmid

Agenda (1/2)

Week	Class Day	Work
1	18/02/2025	• **Lecture: Dynamic Logic** • Exercise on dynamic logic
2	25/02/2025	• **Lecture: Fast Adders** • Exercise on dynamic logic (cont.)
3	04/03/2025	• **Lecture: Project Assignment** • G-P Generation Block • 4 Bit Carry Select Adder (Schematics/Simulation)
4	11/03/2025	• Carry-Merge Unit (atomic element) • Carry Merge Block (Schematic/Simulations)
5	18/03/2025	• D Flip Flop Design (Schematic/Simulations) • Top-level: Start Combining all Blocks
6	25/03/2025	• **Lecture: Floorplan and Layout** • Top-level: Combining all Blocks & Clock Tree
7	01/04/2025	• Optimization of the overall design (transistor sizing) • Toplevel verification with provided tester

The schedule is tight, but manageable.
Please attend every week in person and ask questions and discuss with the TAs

Agenda (2/2)

Week	Class Day	Work
8	08/04/2025	• **Midterm Presentation (graded): toplevel verified (frontend only)** • Initial high-level floorplan
9	15/04/2025	• Discussion and Refinement of Floorplan with the TAs • **Check point: floorplan completed** • Start with Layout
10	22/04/2025	EASTER VACATION
11	29/04/2025	• Drawing the layouts
12	06/05/2025	• Drawing the layouts
13	13/05/2025	• Drawing the layouts
14	20/05/2025	• Drawing the layouts
15	27/05/2025	• Finalizing the layouts • **Final Presentation (graded): verified post-layout**

The schedule is tight, but manageable.
Please attend every week in person and ask questions and discuss with the TAs

Evaluation

Check points

- Assistants will be checking the design status of your blocks/design
- Date: April 15th
- Mandatory, but not graded: it is YOUR responsibility to make sure you have this discussion with a TA

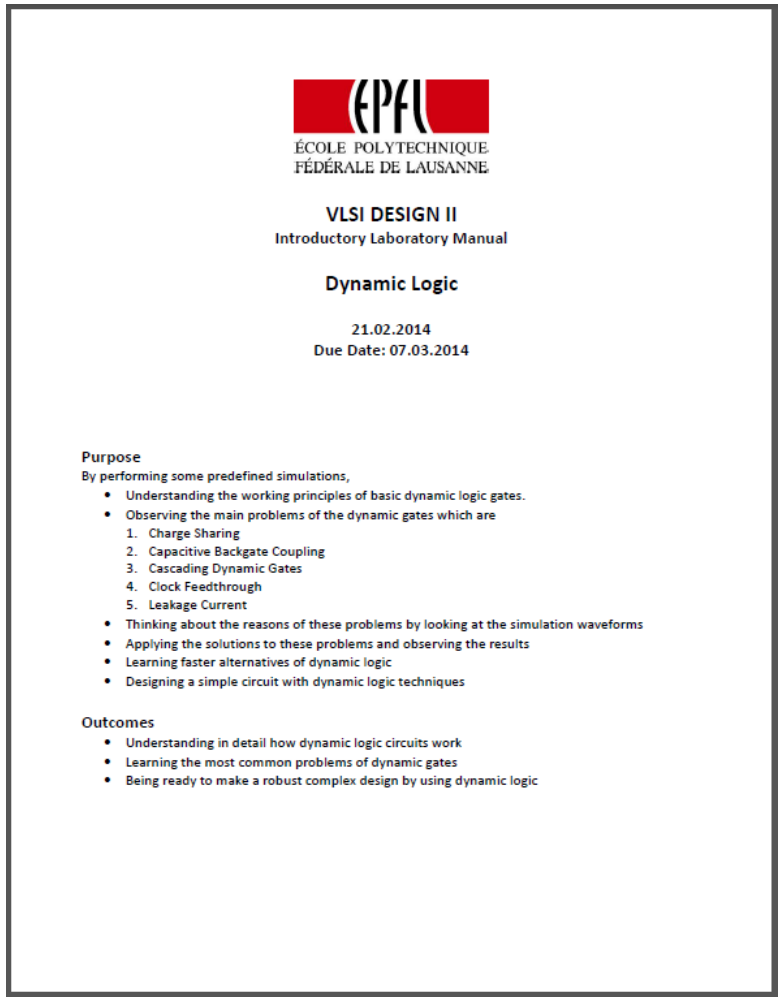
Presentations

1. **Intermediate**
   - Date: April 8th
   - Content: verified top-level schematic
2. **Final**
   - Date: final week of the semester (Tuesday during class or later in the week)
   - Content: verified post-layout top-level

Grading

- **50/50 based on intermediate & final presentations**

First Step: Introductory Laboratory



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

VLSI DESIGN II
Introductory Laboratory Manual

Dynamic Logic

21.02.2014
Due Date: 07.03.2014

Purpose

By performing some predefined simulations,

- Understanding the working principles of basic dynamic logic gates.
- Observing the main problems of the dynamic gates which are
  1. Charge Sharing
  2. Capacitive Backgate Coupling
  3. Cascading Dynamic Gates
  4. Clock Feedthrough
  5. Leakage Current
- Thinking about the reasons of these problems by looking at the simulation waveforms
- Applying the solutions to these problems and observing the results
- Learning faster alternatives of dynamic logic
- Designing a simple circuit with dynamic logic techniques

Outcomes

- Understanding in detail how dynamic logic circuits work
- Learning the most common problems of dynamic gates
- Being ready to make a robust complex design by using dynamic logic

1. Problems of Dynamic Gates

- Simulate
- Try to understand the problem
- Learn the reasons of the problem
- Apply the solution
- Compare the results

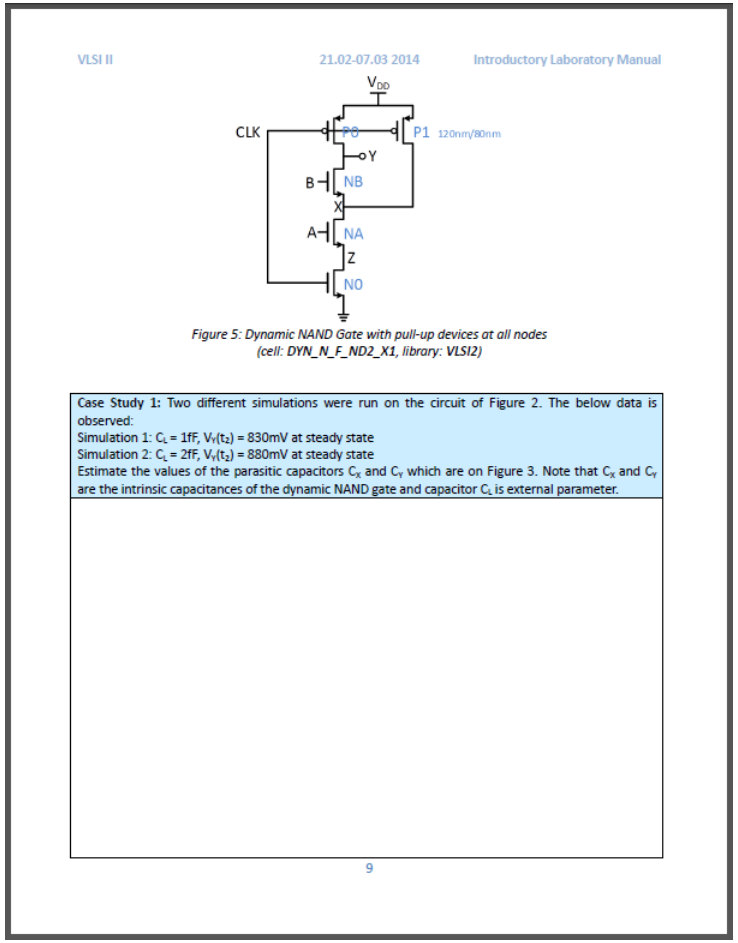
2. Faster Dynamic Logic Solutions

- Understand the idea for increasing the speed
- Simulate
- Compare

3. A Small Design Example

- Apply what you have learned to design a small combinational circuit

First Step: Introductory Laboratory



VLSI II 21.02-07.03 2014 Introductory Laboratory Manual

Figure 5: Dynamic NAND Gate with pull-up devices at all nodes (cell: DYN_N_F_ND2_X1, library: VLSI2)

Case Study 1: Two different simulations were run on the circuit of Figure 2. The below data is observed:
Simulation 1: C_L = 1fF, $V_Y(t_2)$ = 830mV at steady state
Simulation 2: C_L = 2fF, $V_Y(t_2)$ = 880mV at steady state
Estimate the values of the parasitic capacitors C_X and C_Y which are on Figure 3. Note that C_X and C_Y are the intrinsic capacitances of the dynamic NAND gate and capacitor C_L is external parameter.

9

- **Case Studies**
  - Based on the simulated circuits
  - To be done after the course
- **Check Points**
  - Only one check point at the end of the whole Manual based on the Small Design Example

Questions?