香 港 中 文 大 學
The Chinese University of Hong Kong

CENG5030

Part 1-4: Switching Activity

Bei Yu

(Latest update: March 25, 2019)

Spring 2019

These slides contain/adapt materials developed by

- Sukumar Jairam et al. (2008). “Clock gating for power optimization in ASIC design cycle theory & practice.”. In: *Proc. ISLPED*, pp. 307–308

- C and A are intertwined
- $P = V^2 \times f \times C_{effective.}$
- ILP + Frequency increase => Power problem!!
- Factors affecting A:
  - Complexity of the processor
  - Exploitation of parallelism
  - Bit-width of its structures etc.
  - Optimized at the architectural and microarchitectural level
  - Can be changed by run-time optimizations
- Factors affecting C:
  - Size of a processor's structure
  - Organization to exploit locality
  - Manipulated at the circuit and process technology level
  - Determined at fixed design time

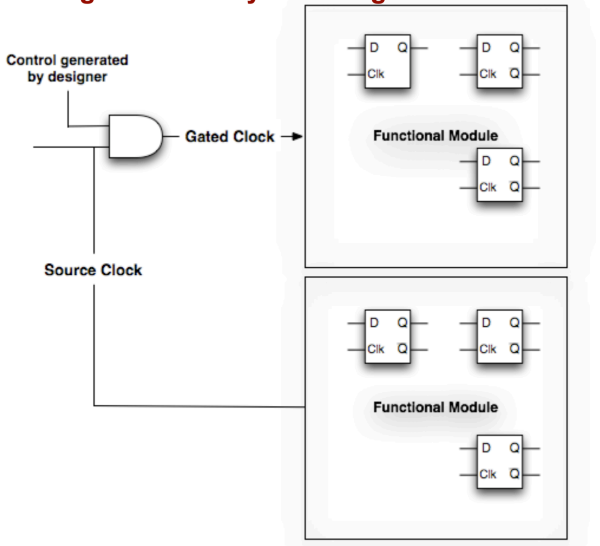
On Switching Activity

- **Idle-Unit switching activity:**
  - Triggered by clock transitions in unused portions of hardware.
- **Idle –width switching activity :**
  - Mismatch in the implemented and the actual width of processor structures.
- **Idle-capacity switching activity :**
  - When a program does not use the provided hardware architectures in their entirety.
- **Parallel switching activity:**
  - Activity expended in parallel for performance
- **Cacheable switching activity:**
  - Repetitive switching activity, convert computing activity to cache lookups
- **Speculative switching activity:**
  - Speculatively executing incorrect instructions is wasted activity
- **Value- dependent switching activity:**
  - Power consumed depends on the actual data values.

Excess Switching Activity	Cause	Granularity	Line of Attack	Example Technique	Section
Idle-unit	Clock-induced switching in unused (idle) units	Functional unit	Clock gating	Clock-gated Functional Units [11, 218, 152, 57, 58]	4.2
Idle-width	Bit-width too wide for typical operations	Cross section of FUs, datapaths, caches	Adapting to narrow-width operands	Clock-gated high-order bits in ALUs [37, 44], cache compression [221, 235, 234, 237, 141]	4.3, 4.4
Idle-capacity	Processor structures sized to support peak ILP not fully utilized in typical programs	Large processor structures: instruction queues, core width, caches	Dynamic resizing	instruction queue resizing [42, 80, 182], cache resizing [244, 8, 21, 68, 9, 168, 241, 131]	4.5,4.6, 4.7, 4.8
Parallel-speculative	Parallel (speculative) activity for speed	Caches, coherence H/W	Serializing or filtering parallel activity	Way prediction and other techniques for set-associative caches [95, 87, 133, 109, 183, 242, 249, 168, 241, 131] Coherence, [171]	4.9
Cacheable (repetitive)	Repetitive computing with the same inputs, or repetitive memory accessing	Architectural structures: FUs, caches	Caching—or memoization	Work reuse [56, 86, 107, 208], filter cache [142], loop buffers [150, 24, 25, 232, 10, 110], trace caches [193, 210]	4.10
Speculative	Activity wasted on wrong speculation	Out-of-order core	Execution throttling	Pipeline & selective gating [161, 16]	4.11
Value-dependent	Data value encoding not optimal	FU, datapaths	Applying different data encoding	Bus encodings [176, 75, 27, 28, 173, 212, 26, 188, 55, 233]	4.12

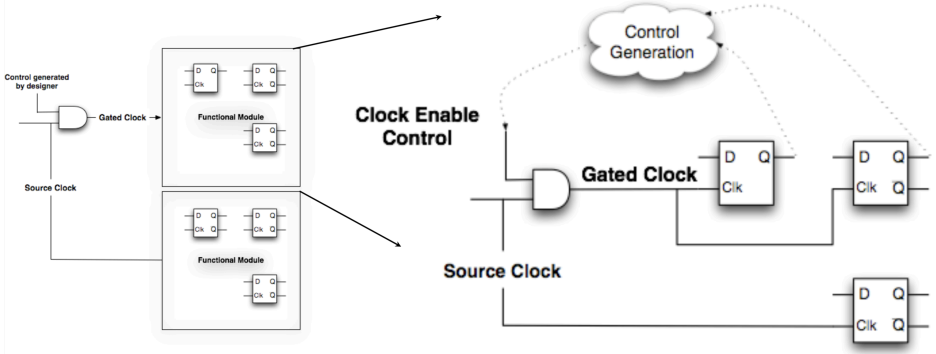
Background: Clock Gating Overview

- **System level gating: Turn off entire block disabling all functionality.**
- **Conditions for disabling identified by the designer**



Background: Clock Gating Overview

- System level gating: Turn off entire block disabling all functionality.
- Conditions for disabling identified by the designer

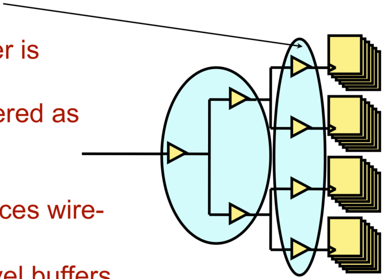


- Suspend clocks selectively
- No change to functionality
- Specific to circuit structure
- Possible to automate gating at RTL or gate-level

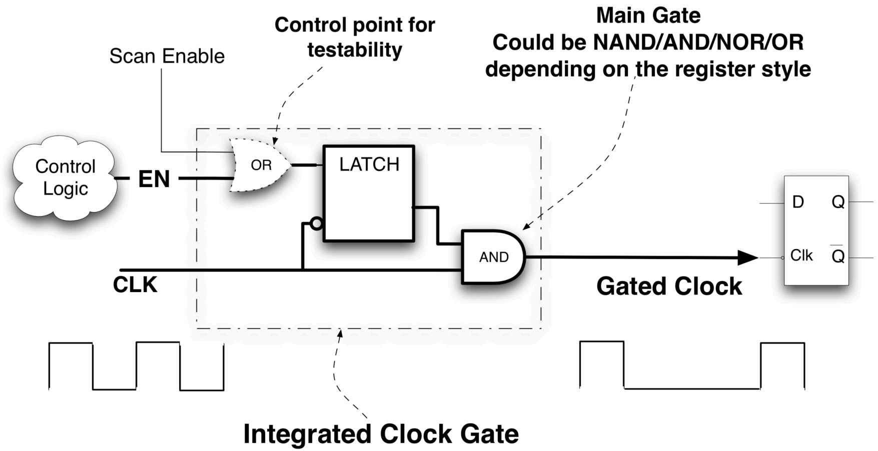
Background: Clock Gating Overview

- Clock network power consists of
  - Clock Tree Buffer Power
  - Clock Tree dynamic power due to wires
  - CLK->Q sequential internal power
- Leaf-levels drive the highest capacitance in the tree
- ~80% of the clock network dynamic power is consumed by the leaf driver stage
  - The clock pins of registers are considered as loads
  - Leaf cap = wire cap + (constant) pin cap
  - Good clustering during synthesis reduces wire-cap
- Effective clock gating isolates this leaf level buffers and cap, providing large dynamic power savings
- Larger savings with CGs higher up in the tree
  - A trade-off with timing

Clock network consumes 30-50% of the total dynamic power of the chip



Background: Clock Gating Overview



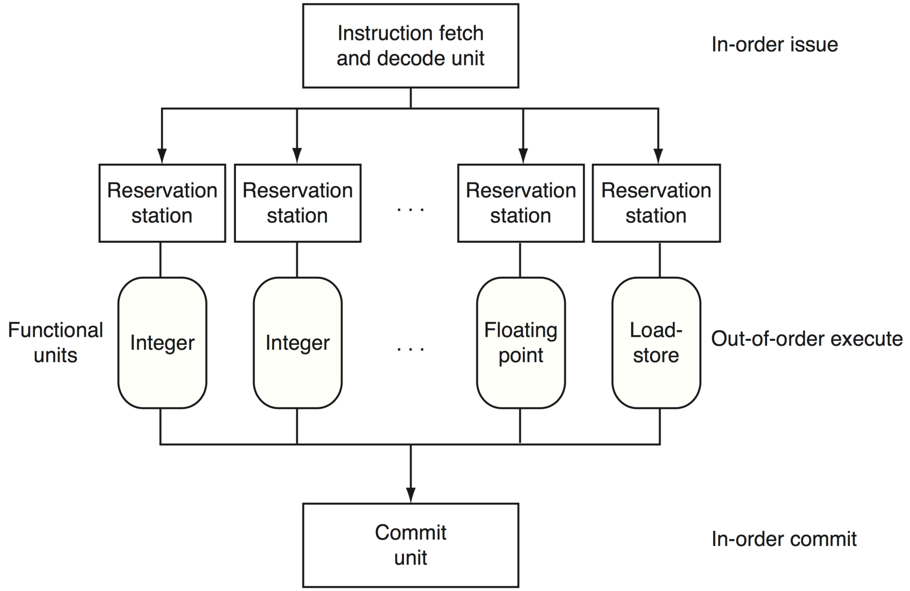
Background: Superscaler

SuperScaler – Dynamic multiple-issue processors

Use hardware at run-time to dynamically decide which instructions to issue and execute simultaneously

- Instruction-fetch and issue – fetch instructions, decode them, and issue them to a FU to await execution
- Defines the Instruction lookahead capability – fetch, decode and issue instructions beyond the current instruction
- Instruction-execution – as soon as the source operands and the FU are ready, the result can be calculated
- Defines the processor lookahead capability – complete execution of issued instructions beyond the current instruction
- Instruction-commit – when it is safe to, write back results to the RegFile or D$ (i.e., change the machine state)

Background: In-Order v.s. Out-of-Order



Switching Activity – Circuit Level[1]

[1] Hai Li et al. (2004). "DCG: deterministic clock-gating for low-power microprocessor design". In: *IEEE TVLSI* 12.3, pp. 245–254.

Background: Instruction Fields

MIPS fields are given names to make them easier to refer to

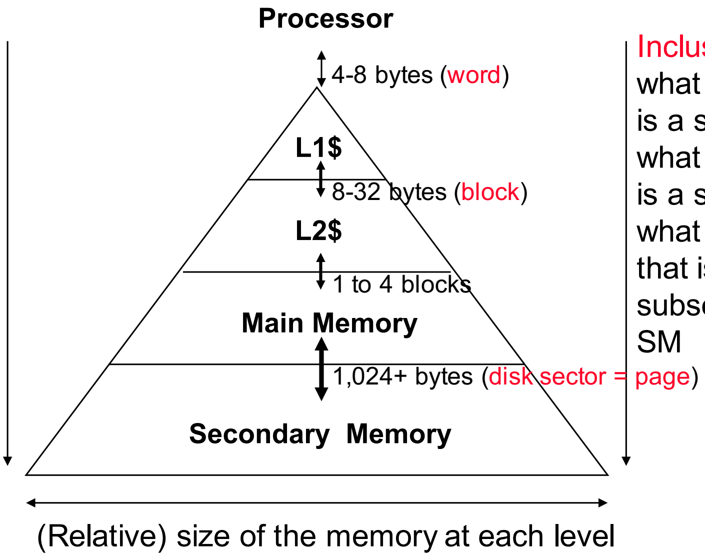
6	5	5	5	5	6
op	rs	rt	rd	shamt	funct

op	6-bits,	opcode that specifies the operation
rs	5-bits,	register file address of the first source operand
rt	5-bits,	register file address of the second source operand
rd	5-bits,	register file address of the result's destination
shamt	5-bits,	shift amount (for shift instructions)
funct	6-bits,	function code augmenting the opcode

Switching Activity – Core[2]

[2] David Brooks and Margaret Martonosi (1999). “Dynamically exploiting narrow width operands to improve processor power and performance”. In: *Proc. HPCA*, pp. 13–22.

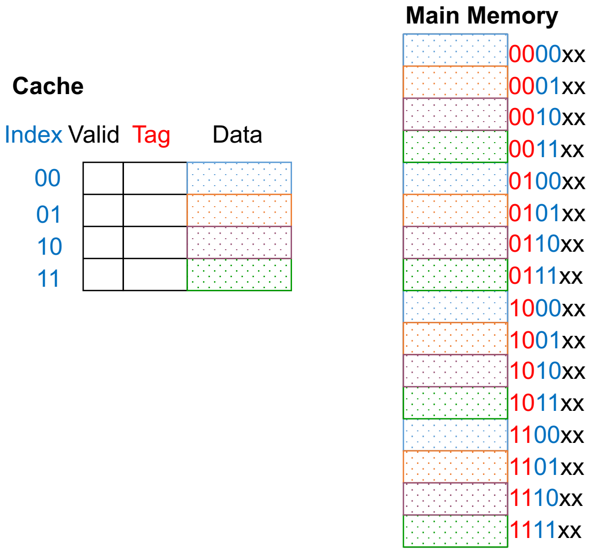
Background: Memory System



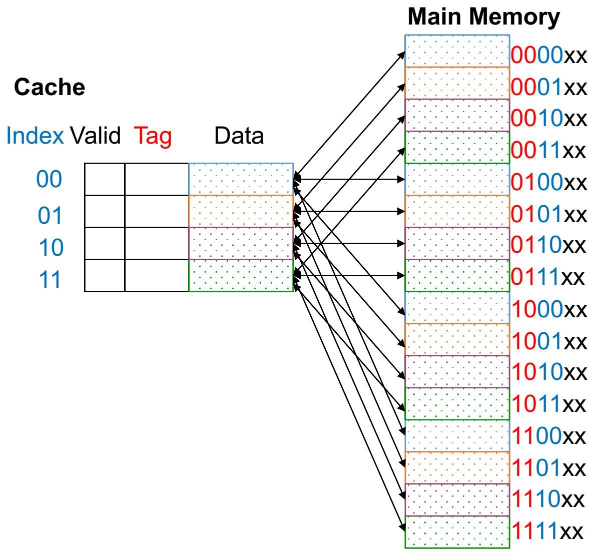
Processor
4-8 bytes (word)
L1$
8-32 bytes (block)
L2$
1 to 4 blocks
Main Memory
1,024+ bytes (disk sector = page)
Secondary Memory
Increasing distance from the processor in access time
(Relative) size of the memory at each level

Inclusive– what is in L1$ is a subset of what is in L2$ is a subset of what is in MM that is a subset of is in SM

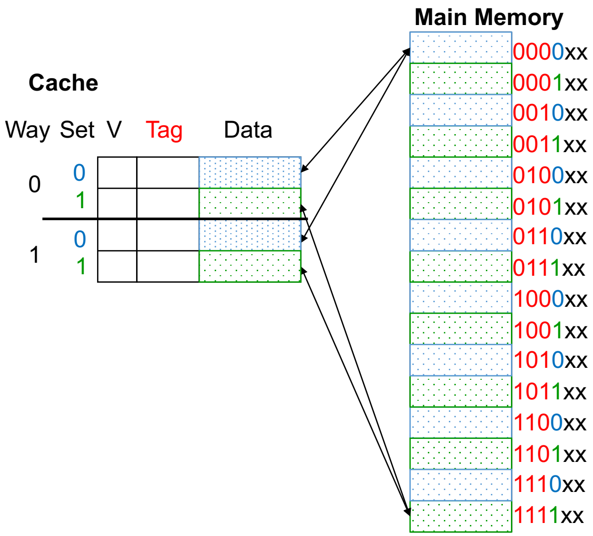
Background: Direct Mapping



Background: Direct Mapping



Background: Set Associative Mapping



Switching Activity – Cache[3]

[3] David H. Albonesi (1999). “Selective cache ways: On-demand cache resource allocation”. In: *Proc. MICRO*, pp. 248–259.