

EE115C: Digital Electronic Circuits

Professor Markovic

Thilan Tran

Spring 2021

Contents

EE115C: Digital Electronic Circuits	2
Scaling Trends	3

EE115C: Digital Electronic Circuits

- design abstraction levels:
 - devices, circuits, gates, modules, systems, chips
 - * each layer encapsulates the previous layers
 - in 115C, examining the gate level and below
- course goals are understanding, designing, and optimizing digital circuits with respect to different metrics
 - eg. power dissipation, speed
- design flow:
 1. schematic
 - Virtuoso
 2. layout
 - Virtuoso
 3. verification
 - Assura database, DRC, LVS
 4. simulations:
 - Spectre models
 - compare results from initial schematic simulations and verification simulations
- history:
 - the Babbage Difference Engine was the first computer from 1832
 - * mechanical system with interlocking gears
 - the Zuse Z3 was the first digital electronic computer from 1941:
 - * used 2k electromechanical relays
 - * binary, 5-10Hz operating speed
 - ENIAC came 5 years later in 1946:
 - * used 18k vacuum tubes, and had 5 million hand-soldered joints
 - * decimal
 - the first PC was Simon from 1950:
 - * used electromechanical relays
 - * supported 4 operations, addition, negate, shift, and store
 - the transistor revolution occurred in the 1940s:
 - * first transistor made in Bell Labs in 1948
 - * built using paperclips
 - the first integrated circuit was created in 1958:
 - * with Texas instruments
 - * multiple transistors mounted on the same silicon germanium
 - * beginning of integrated electronics
 - Intel made the 4004 microprocessor in 1971:
 - * 108kHz processor

- * 2300 transistors
 - shows the power of integration when compared with the massive Z3
- Moore's Law from 1965:
 - * predicted that every two to three years, the number of transistors on a chip doubled
 - * has mostly held true
 - * from 10 micrometer technology in 1972 to 20 nanometer technology in 2012

Scaling Trends

- scaling trends:
 - historically, there were different approaches to scaling
 - initially constant voltage scaling, then constant E-field scaling, then general scaling
 - want to reduce voltages and sizes to fit even more transistors together, while using less power:
 - * V_{DD}, V_T
 - * W, L, t_{ox} where t_{ox} is the oxide thickness
 - Dennard's paper on classical MOSFET scaling in 1974:
 - * defines how voltage, current, capacitance, length all scale down together
 - * shows how scaling different parameters affects the remaining parameters
- in the earliest stages, there was no concern about power usage, so constant voltage scaling was performed:
 - ie. constant V_{DD}, V_T and sizes scaled by $\frac{1}{S}$
 - BJT was initially used, and then switched to nMOS in the 1980s and later CMOS in the 1990s
 - * CMOS was more energy efficient and improved the integration level
 - with constant voltage, power density explodes at a factor of $\frac{1}{S^2}$, leading to difficulty cooling
- in the 1990s, the issue with power density was dealt with by using constant E-field scaling:
 - ie. both size and voltage parameters are scaled by $\frac{1}{S}$
 - * thus $E = \frac{V}{L}$ is a constant
 - pros:
 - * more transistors per area $\frac{1}{S^2}$
 - * faster delay $\frac{1}{S}$

- * lower energy per operation $\frac{1}{S^3}$
- cons:
 - * unavoidable exponential leakage AKA V_T scaling
- some solutions to perform leakage control in a bleeding blood vessel analogy:
 - * “*pinch*” the vessel against the bone ie. stop the flow
 - strategy used in **fully-depleted silicon on insulator (FD-SOI)**
 - * using fins ie. pinch the vessel itself
- thus constant E-field scaling ended at the 130 nm node due to issues with exponential leakage:
 - instead use general voltage scaling where we scale geometry more aggressively than the voltage:
 - * ie. size parameters are scaled by $\frac{1}{S}$ while voltage parameters are scaled by $\frac{1}{U}$
 - * generally, voltage scaling is slowing down such that $S > U$
 - * V_T and t_{ox} are set by leakage constraints
 - this lead to a need to do parallelism in a multicore revolution to achieve greater performance:
 - * clock speeds stop to scale
 - * use multiple parallel cores, but don’t use all of them to maximize technology
 - * AKA underutilizing silicon due to power limitations
 - fraction of silicon at full capability, or full silicon at fraction of capability
- technology scaling is power driven
 - have to switch technologies when we hit a power wall
- more than Moore:
 - to continue following Moore’s law, need to continue to *miniaturize*:
 - * already approaching atomic limits
 - * 7nm transistors fins are only 25 atoms wide
 - * transistor scaling is projected to end at 3nm
 - instead, we need to move beyond Moore and consider *diversification*:
 - * technology scaling is reaching its limits, so design becomes so much more important
 - alternate technologies like quantum computing have their own limitations
 - * custom silicon is 1000x faster than general purpose silicon