Silicon Showdown - C 2027

1. **DESCRIPTION**: Participants will be assessed on their understanding of concepts of computer systems, architecture, and microarchitecture.
2. **A TEAM OF UP TO**: 2 **APPROXIMATE TIME**: 50 minutes
3. **EVENT PARAMETERS**:
    a. Each team may bring one three-ring binder of any size containing information in any form and from any source attached using the available rings. Sheet protectors, lamination, tabs and labels are permitted. Participants may remove information or pages for their use during any part of the event.
4. **SCORING**:
    1.1. Highest score wins.
    1.2. Selected questions may be used as tiebreakers.
5. **THE COMPETITION**:
    a. The exam should span across the topics listed below with adequate breadth.
        i. Furthermore, questions should emphasize conceptual understanding and key trends in and between the fields, avoiding memorized formulas and rote arithmetic.
        ii. Subject matter underpinning the topics below but not explicitly listed should not be directly tested but is fine to expect as background knowledge for application questions. For example, questions should not have competitors differentiate between different kinds of logic gates, but that knowledge would be reasonable to expect a competitor to have.
        iii. A written exam would be the natural format of this event, but lab portions are possible as well. If an ES desires to run a lab portion dependent on specific software (for example), they must have adequate resources to be able to run the lab (e.g. a way to make the software accessible to the competitors easily) and publish details in order to facilitate viability for competitors at least 2 weeks in advance.
    b. History section topics:
        i. History of chip design limited to:
            1. Foundations: Fairchild Semiconductor, DEC, IBM, Texas Instruments, Intel, Motorola, Sun Microsystems, MIPS
            2. Infrastructure: Cadence, Synopsys, TSMC, ASML
            3. Current players: Apple, AMD, Arm, Broadcom, Qualcomm, NVIDIA, MediaTek
            4. Specialized: Tesla, Google
            5. Newer players: Cerebras, Groq, Etched, Tenstorrent, SiFive
        ii. Questions pertaining to this section should not emphasize factoids like specific dates when events occurred, instead key relationships between companies and the evolution of the field
    c. Computer systems section topics:
        i. Processes and threads: the process vs thread distinction, address space layout, context switching, and scheduling algorithms, emphasizing the trade-offs between them (fairness, throughput, latency, starvation) rather than memorizing any one algorithm
        ii. Concurrency and synchronization: race conditions and critical sections, the conceptual roles of mutexes, semaphores, and condition variables, atomicity, and deadlock (the four necessary conditions and how breaking one prevents it)

iii. Inter-process communication, limited to: pipes, signals, shared memory, message passing, and sockets
iv. POSIX system calls, limited to: open(), read(), write(), close(), fork(), execvp(), wait(), and pipe(), emphasizing what abstraction each call provides and how they compose (e.g. fork/exec/wait, or pipe across processes), not exact signatures or flag values
v. File systems and I/O: the file descriptor abstraction, buffering and caching, block vs character devices, and persistence
vi. Communication protocols, limited to RS-232, TCP/IP, UDP, UART, SPI, I2C, CAN, and Ethernet, compared along conceptual axes:
    1. Serial vs parallel; synchronous vs asynchronous
    2. Topology and addressing: point-to-point vs bus vs multi-master
    3. Reliability, ordering, and flow-control guarantees
    4. Layers
vii. Data encodings limited to FM and MFM
viii. Error detection and correction, limited to parity, checksums, CRC, ECC, and Hamming codes
ix. Real-time systems: hard vs soft real-time, determinism, priority inversion

d. Computer architecture/microarchitecture section topics:
i. Combinational vs sequential logic design
ii. Finite state automata, datapaths, microcode
iii. Instruction Set Architectures (Reduced (RISC) vs Complex (CISC), very long instruction word (VLIW)), microarchitecture. Do not emphasize obscure details
iv. Amdahl's Law, Moore's Law, Instructions per Cycle (IPC) vs Cycles per Instruction (CPI) and pitfalls of these metrics
v. Trade-offs and design principles - critical path, balance, throughput, latency
vi. Processes: privilege vs priority, interrupts and exceptions
vii. Microcontrollers vs microprocessors, datasheet analysis
viii. Field programmable gated arrays (FPGA) vs Application specific integrated circuits (ASIC)
ix. Computer arithmetic: adders, multipliers (Do not test details of specific algorithms)
x. Memory hierarchy, physical memory (random access memories (SRAM, DRAM), interleaving, unalignment), virtual memory and translation lookaside buffers, cache memory, non uniform memory access (NUMA) vs uniform memory (UMA)
xi. Specialization: heterogeneous architectures, GPUs, other accelerators
xii. Speculation, prediction, and performance improvements: branch prediction, cache replacement policies, prefetching
xiii. Parallelism: pipelines and hazards, In-order vs Out-of-order execution (Tomasulo's algorithm, register alias table, reorder buffer, physical registers), superscalar/vector architectures, multicore and cache coherence, load store queue, Flynn's Taxonomy, interconnection types

6. **NOTES FOR EVENT SUPERVISORS**:
a. For sufficiently complex topics and/or questions, it would be appropriate to have more stimulus styled problems emphasizing critical thinking with the background knowledge provided rather than requiring extensive knowledge on the subject

7. **ADVICE AND SUGGESTED RESOURCES:**

7.1. Some topics that are advisable to have a foundational background in to succeed in this event include: digital logic design, first principles of circuit theory, binary and hexadecimal representations of numbers, and data structures.
7.2. Turing Complete (educational video game) to develop some of the digital logic background
7.3. Dr. Onur Mutlu's YouTube channel has several playlists full of lectures which are great resources. The Computerphile YouTube channel is also a fantastic resource.
https://www.youtube.com/onurmutlulectures
https://www.youtube.com/user/Computerphile
7.4. https://www.unix.com/man_pages/posix
7.5. Recommended textbooks would include Computer Architecture: A Quantitative Approach and Operating Systems: Principles and Practice

8. **SAMPLE QUESTIONS:**

8.1. What makes the TSMC-ASML partnership so difficult for competitors to disrupt?
A) The two companies share a restrictive intellectual property agreement where any process improvement discovered by TSMC is legally barred from being applied to ASML machines sold to Intel or Samsung.
B) TSMC's massive volume of production provides ASML with immense amounts of data and defect metrics, which ASML uses to perfect their machines for future generations.
C) ASML relies exclusively on TSMC to manufacture the custom AI accelerators required to run the complex computational lithography software inside the EUV machines, creating mutual dependency.
D) TSMC grants ASML exclusive rights to its proprietary photomask designs, allowing ASML to physically lock its machine calibrations to TSMC's specific manufacturing architecture.
8.2. How are child processes differentiated from the parent process following a fork()?
8.3. Consider an out-of-order CPU executing the following instructions, where the pipeline stages consist of fetch, decode, register rename, execution, and writeback. There is no data forwarding. Furthermore, add instructions take 1 cycle during execution and multiply instructions take 3 cycles during execution. Assume there is no shortage in the reservation stations and functional units. Writeback can handle one instruction at a time, and prioritizes older instructions. Fill in the missing blocks in the processor trace. Represent stalls with a "-"
ADD R0, R1, R1
MUL R2, R0, R0
MUL R3, R3, R3