# Indian Institute of Technology Tirupati
# Computer System Architecture (CS5202)

**Syllabus for End-semester**

Jan-May 2020

---

1. **Cache memory system:** Introductory concepts of cache memory, Cache performance, Four memory hierarchy questions, Six basic cache optimization methodologies, Ten advanced cache performance optimizations.

   Reading Material: Appendix B of [1]: Section B.1 (includes Opteron Data Cache), B.2, and B.3, ; Case studies 1 and Case studies 2. Chapter 2 of [1]: Section 2.1, 2.2 and Case study 1: Optimizing cache performance via advanced techniques. Lectures[3]: Lecture 7, 8, 9, and 10.

2. **Main memory system:** Memory cell, DRAM technology, Main memory hierarchy, Memory array, Banking, Chip, Rank, Dual inline memory module, Memory channel, Memory interleaving and Address mapping.

   Reading Material: Chapter 10 of [4]: Section 10.1 to 10.3, Chapter 13 of [4]: Section 13.3. Chapter 2 of [1]: Section 2.3. Lectures: Lecture 11 of class [3]; (Additional materials from [5]: Lecture 1, Lecture 2, and Lecture 3)

3. **Virtual memory system:** Introductory concepts of virtual memory, four memory hierarchy questions, virtual to physical address translation, virtually/physically indexed and tagged cache, AMD opteron TLB and cache architecture, paged virtual memory (AMD Memory management), ARM cortex A53 virtual memory, (additional reading: Intel i7 and A8 memory hierarchy).

   Reading Material: Appendix B of [1]: Section B.4, Section B.5 (You may read only Paged virtual memory, Protection will not be included in the test); Chapter 2 of [1]: Section 2.6; Lecture: Lecture 12 from [3].

4. **Pipeline architecture:** Instruction level parallelism, CPU design, Single- and Multi-cycle design, The five stage pipeline architecture, Pipeline performance, Pipeline processor design (Pipeline implementation), Hazards and dependencies, Long latency pipeline and multiple functional units, Dependencies/Hazards in long latency pipeline, Hazards mitigation: branch predictors, stall cycle insertion, forwarding, compiler based solution, Exception handling in pipeline (this will not be part of test), Pipeline performance considering stalls.

   Reading Material: Appendix C of [1]: C.1, C.2, C.3, C.4, C.5; Chapter 4 of [6]: Section 4.3, 4.4, 4.5, 4.6, 4.7, 4.8; Lectures from [3]: Lecture 13.1 and 13.2.

5. **Superscalar architecture:** Instruction level parallelism: Challenges, Limits of pipeline, Amdhal's law, Classes of ILP machines: super pipelined, super-scalar pipeline, VLIW pipeline, Loop unrolling, software pipelining, Super-scalar architecture concepts (wider pipes, diversified and specialised units, out-of-order execution), Superscalar stages, Instruction flow, Fetch alignment, I-Cache Organisation of IBM-RS/6000, Buffers, CISC based Decoding, Pre-decoding in AMD, Dispatch and Issue, Reservation station, challenges in Execution stage, Completion and Retirement, Handling control flow, Classes of Branch instructions, Control-flow graph, Disruption due to control flow, Branch prediction (Condition speculation and Target speculation), Branch Target Buffer with 2-bit predictor, Recovery from miss prediction, Advanced predictors (Correlating predictor, Two-level prediction, gshare, Tournament, Tagged hydbrid predictor, Register data flow, Renaming, Tomasulo's Algorithm, Reorder buffer, Memory data flow, Memory access ordering, Load bypassing, and Load forwarding.

   Reading Material: Chapter 4 of [2]: Section 4.1, 4.2, and 4.3; Chapter 5 of [2]: Section 5.1, 5.2, and 5.3; Chapter 3 of [1]: Section 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8 and 3.9; Lectures of [3]: Lecture 14, 15, 16 and 17.

# References

[1] Henessy and Patterson, Computer Architecture Quantitative Approach, $5^{th}$ Edition.

[2] John P. Shen and Mikko H. Lipasti, Modern Microprocessor Design - Fundamentals of Superscalar Processors.

[3] Course lecture: Link.

[4] Bruce Jacob, Spencer Ng, and David Wang; Memory Systems: Cache, DRAM, Disk; 2008, Elsevier.

[5] Onur Mutlu, Scalable Memory System Lectures, Link

[6] Patterson and Henessy, Computer Organisation and Design: A Hardware/Software Interface, $5^{th}$ Edition.