Real-Time Expert Systems Computer Architecture
Algorithm-Architecture Matching for Signal and Image Processing
Readings in Computer Architecture
Multiscalar Processors
Parallel Machines: Parallel Machine Languages
Advanced Computer Architectures
Processor Architecture
The Architecture of Pipelined Computers
Essentials of Computer Organization and Architecture
Implementation of a General-purpose Dataflow Multiprocessor
Algorithms & Architectures For Parallel Processing, 4th Intl Conf
Cloud Computing
The Fifth Generation Computer Project
Source Code Optimization Techniques for Data Flow Dominated Embedded Software
Advanced Topics in Dataflow Computing and Multithreading
Handbook of Signal Processing Systems
DataFlow Supercomputing Essentials
Dataflow Architecture for Machine Control
Design of Image Processing Embedded Systems Using Multidimensional Data Flow
Network Processor Design
Stream Processor Architecture
Advanced Computer Architecture
Advanced Topics in Dataflow Computing and Multithreading
Processor Microarchitecture
Bio-Inspired Models of Network, Information, and Computing Systems
A Processor Architecture for Dynamic Memory Disambiguation
Computer Architecture
Parallel and Distributed Processing and Applications
Algorithms & Architectures
Algorithms and Architectures for Real-Time Control 1992
Parallel Processing and Parallel Algorithms
Multithreading Architecture
Multi-Processor System-on-Chip 2
Modern Processor Design

LLOYD BRAY
This book presents a coherent approach to computer system design that encompasses many, if not most, of the design problems and solutions.
Multiscalar Processors presents a comprehensive treatment of the basic principles of Multiscalar execution, and advanced techniques for
Multithreading System-on-Chip 2
Modern Processor Design

This Workshop focuses on such issues as control algorithms which are suitable for real-time use, computer architectures which are suitable for real-
time control algorithms, and applications for real-time control issues in the areas of parallel algorithms, multiprocessor systems, neural networks,
fault-tolerance systems, real-time robot control identification, real-time filtering algorithms, control algorithms, fuzzy control, adaptive and self-tuning
control, and real-time control applications.
Advanced Topics in Dataflow Computing and Multithreading Springer Science & Business Media

This lecture presents a study of the microarchitecture of contemporary microprocessors. The focus is on implementation aspects, with discussions on their implications in terms of performance, power, and cost of state-of-the-art designs. The lecture starts with an overview of the different types of microprocessors and a review of the microarchitecture of cache memories. Then, it describes the implementation of the fetch unit, where special emphasis is made on the required support for branch prediction. The next section is devoted to instruction decode with special focus on the particular support to decoding x86 instructions. The next chapter presents the allocation stage and pays special attention to the implementation of register renaming. Afterward, the issue stage is studied. Here, the logic to implement out-of-order issue for both memory and non-memory instructions is thoroughly described. The following chapter focuses on the instruction execution and describes the different functional units that can be found in contemporary microprocessors, as well as the implementation of the bypass network, which has an important impact on the performance. Finally, the lecture concludes with the commit stage, where it describes how the architectural state is updated and recovered in case of exceptions or mispeculations. This lecture is intended for an advanced course on computer architecture, suitable for graduate students or senior undergrads who want to specialize in the area of computer architecture. It is also intended for practitioners in the industry in the area of microprocessor design. The book assumes that the reader is familiar with the main concepts regarding pipelining, out-of-order execution, cache memories, and virtual memory.

Multithreaded Computer Architecture: A Summary of the State of the ART Springer Science & Business Media

Table of Contents: Introduction / Caches / The Instruction Fetch Unit / Decode / Allocation / The Issue Stage / Execute / The Commit Stage / References / Author Biographies

Media processing applications, such as three-dimensional graphics, video compression, and image processing, currently demand 10-100 billion operations per second of sustained computation. Fortunately, hundreds of arithmetic units can easily fit on a modestly sized 1cm² chip in modern VLSI. The challenge is to provide these arithmetic units with enough data to enable them to meet the computation demands of media processing applications. Conventional storage hierarchies, which frequently include caches, are unable to bridge the data bandwidth gap between modern DRAM and tens to hundreds of arithmetic units. A data bandwidth hierarchy, however, can bridge this gap by scaling the provided bandwidth across the levels of the storage hierarchy. The stream programming model enables media processing applications to exploit a data bandwidth hierarchy effectively. Media processing applications can naturally be expressed as a sequence of computation kernels that operate on data streams. This programming model exposes the locality and concurrency inherent in these applications and enables them to be mapped efficiently to the data bandwidth hierarchy. Stream programs are able to utilize inexpensive local data bandwidth when possible and consume expensive global data bandwidth only when necessary. Stream Processor Architecture presents the architecture of the Imagine streaming media processor, which delivers a peak performance of 20 billion floating-point operations per second. Imagine efficiently supports 48 arithmetic units with a three-tiered data bandwidth hierarchy. At the base of the hierarchy, the streaming memory system employs memory access scheduling to maximize the sustained bandwidth of external DRAM. At the center of the hierarchy, the global stream register file enables streams of data to be recirculated directly from one computation kernel to the next without returning data to memory. Finally, local distributed register files that directly feed the arithmetic units enable temporary data to be stored locally so that it does not need to consume costly global register bandwidth. The bandwidth hierarchy enables Imagine to achieve up to 96% of the performance of a stream processor with infinite bandwidth from memory and the global register file.