14-10-2024

System Architecture

Ashish Kuvelkar

- IBM -> PC -> open architecture
  1981
- x86 -> 8086

-> bit serial : only one bit at a time.

-> core memory

-> telle type (before raster display)

CP/m -> control process monitor

-> 8086 vs 8088

-> now other way | 8 bit peripherals, memory costly

-> floating point arithmetic co-processor. 8087

IEEE 756
mantisa & exponent
accuracy.

15-10-2024 PC-XT

CPU - interface -> memory, data, i/o

8284 - clk
8259 - interrupt
8288 - bus controller.

74LS 135 - decoder

8237 - DMA

Sector of hard disk -> 512 bytes

I/o data to (RAM) -> DMA -> hold -> CPU
Memory   hold aknoledge <- CPU

8255 -> additional i/o

8253 -> Timer for delay count

PC-AT

80486

PCI (parallel) ISA

2 interrupt controller

but daichained so cpu receive only one interrupt

80386 -> IA-32
tillnow compatable

I/o mapping   00F -> 3FF

IRQ

Memory mapping

-> Blocks of 64k

FFFF -> ROM BIOS

FFFF0h -> 8086

0000h -> PIC

PC chipset

AGP (or) PCI-E

16-10-2024

PC - parameter.

8086/88 → features → segmented memory (paging 64k)
20 lane
x16 with internal reg.

80286 - 16 bit
↳ real, protected mode
→ multiuser configuration

80386 - 32 bit
↳ paging
burst read/write

Pentium
Superscalar, Branch prediction

Processor design

μP → Instruction set processor → (ISA) → Dynamic static interface

CISC, VLIW (EPIC), RISC

17-10-2024

CISC

Instruction execution → mem → alu →

Features of CISC
- variable instruction length
- widely differing execution times.
- ALL micro coded → μ code mem → μ ops

RISC

CPU design simplified CPU, hard wired, (not micro coded)

LOAD, STORE → not directly manipulated in memory
only from registers.

Superscalar → multiple execution unit.

↳ Execute cycle

$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instruction}}{\text{program}}$$

(CISC ↑, RISC ↓, CISC ↓, RISC ↑)

Instruction level parallel processors

Pipelining, superscalar, VLIW

→ Pentium Pro with Risc core

Super scalar processing

↳ Issue policy { ↳ Register renaming (Data dependencies)

21-10-2024

Speculative execution, Parallel Execution, branch prediction.

RAW, WAW

ROB → sequencial consistency model ↳ 2 Types

weak strong

Control transfer instruction Branch → breaker pipeline

22-10-2024

Pentium II, Pentium III, Pentium IV, 20 pipeline

Micro-architecture

→ netburst

→ core

core 2, i5

46nm

↓

14nm 2015

AMD64 VLIW processor.

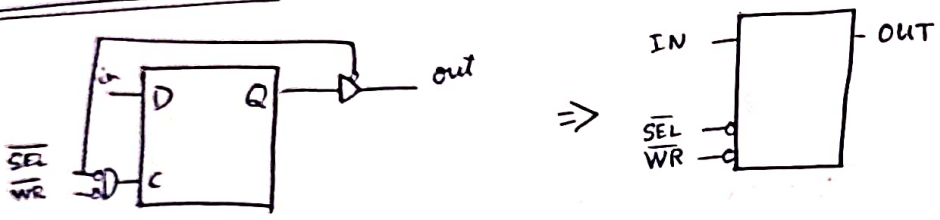
Memory Architecture

use of memory
→ Temporary storage
→ Break the problem in time domain

ROM
- EEPROM
  → hot e⁻ injection

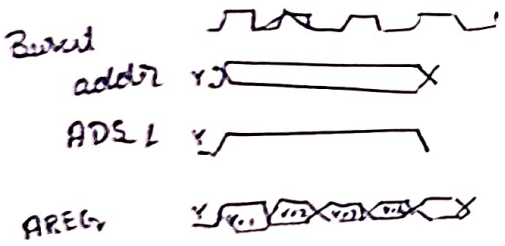
24-10-2024

Structure of SRAM



SRAM array
Read, write timing diagram.

Role of sync RAM over async RAM

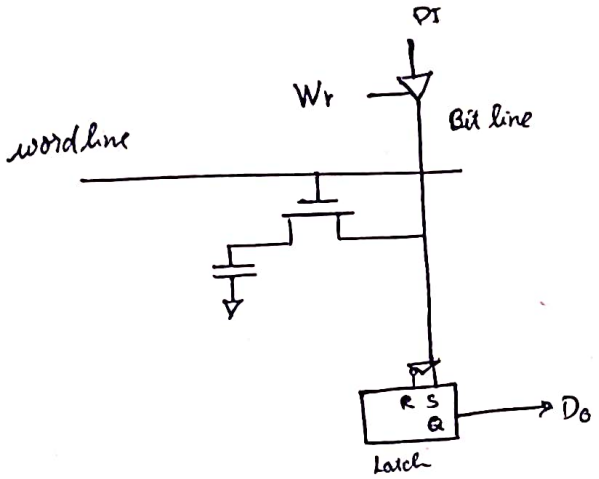


→ Pipelined output.
- one clk delay
- stable o/p

→ High bandwidth. → in GPU

Quad DR → in network switch
→ 4 word of data per clock.

DRAM



1.
2.
3.
4.
5. write back.

→ Symmetric read/write

26-10-2024

ECC memory → Error correction. [ability to correct along with detection]

Rambus → [proprietary tech used for high bandwidth applications

SRAM working. → data sent in highly packed packets

↳ command mode.

DDR2

SSTL_2 & Strobe based data bus.

→ divided to 16 pins & is synchronized with a strobe

168 pin 64 (non) ECC
72 ECC

- DDR-2 → change in prefetch

Cache

ashisch@cdac.in

USB system arch - Don Anderson (hardware)
USB complete. - Jan Axelson (software)

USB

→ No IEEE standard, but .org industrial standard.

- Host ↔ controller (device address)
- Hubs ⇒ to multiply port.

USB endpoint → starting with 0 [all are unidirectional except (endpoint 0)]
↳ control / status PORT

USB enumeration. → them addressed (unique) given

packet { Token, Data, Handshake }

pipe → to carry out communication as packets.
↙ ↘
Stream Message

→ default pipe ⇒ endpoint 0 → USB system software (if present)
↓
device driver

PID → 8 bit
4 bit + 4 bit
↓
so not included in CRC

Transfer modes
control transfer, (scheduled) Isochronous, generated by pooling Interrupt, Bulk transfer (IN, OUT)

control transfer
- init
- enumeration

Isochronous
- reserved
- audio/video
- bth, wifi, ble

Interrupt
keypress
mouse move.
printer

Bulk transfer
↳ storage, wifi, ble.

USB traffic → divided into frame of 1ms

1ms [start of frame | data 1 endpoint 2 | : | :]

1ms [start of frame | device 2 endpoint | : | :]

NRZI encoding | Bit stuffing ⇒ after 6 ones an 0 is added.
↳ Inverted.

Sync pattern 0x80

NAK, STALL — error
↘ buffer full.

data always send as DATA0/DATA1

ERROR 18 bit time → no ack

19-10-2024

CMOS to System

- Integration of components => SoC
- Analog ⇔ PHY layer ↔ Digital
- Software functionality => hardware implementation -> Improves performance.
- Use of Foundation IP
  - Memory, FIFO, Arbiter, FSM
- GATES (Digital, Analog)
  - CMOS gate
  - PLL
  - SAR ADC
- Analog -> Design flow

- ASIC design flow

  RTL → GDS

KPI - Area, freq., power, security

23-10-2024

Formal Verification.

Avani Langhate

-> Frontend

Backend
-> CAD tools.

->

-> Formal verification.

Design verification method.

- Formal verification
  - -> math proof & algorithm
- Simulation
  - -> Behavioral -> Functionality
  - -> RTL -> RTL level
  - -> Gate level ->
- Emulation
  - -> mimic orginal silicon
- FPGA prototyping

overlap, non overlap.
↓
same cycle.

|->
↓
#1 different delay.

Need for formal verification

Intel pentium bug.

Diff between formal & simulation

<antecedent> |=> <consequent>
-> implication operator

-> assertion in FIFO

-> cover

Cadence tool JASPER

- o
- ✓ - Full proof
- ? - undetermined
- x - counter example.

FSV, SPV } safety & security (Automobile se