rived from identifiers explicitly

or implicitely specified in an object program.

Physical or Real addresses OR identifies the fixed physical storage locations in each memory unit M_i.

mapping $f: V \rightarrow R$ is the Key to successful design of a multilevel memory.

Address translation :

Virtual address to Real address.

Translation look aside buffer

Fig 6.26) ⤴

Memory Allocation:

Non Preemptive allocation:

1. First fit ($n_j > n_i$ first occurance)
2. Best fit ($n_j - n_i$ is minimized)

} unpaged system.

Fig 6.34)

Preemptive allocation: (Reallocation)

1. Compaction (Fig-6.35)
2. Replacement policy(Fig 6.36)

Handwired Control

d Design method

method 1:

* Classical method of sequential circuit
* Minimize the amount of hardware in particular by using only $[\log_2 P]$ flipflops.

Method 2:

* 1 flipflop per state
* Known as one-hot-method
* Expensive in terms of flipflops
* Simplifies CU design and debugging.

State table

-fig 5.4

$$\left.\begin{array}{l}\text{* mealy machine}\\ \text{* moore machine}\end{array}\right\} \rightarrow \text{finite state machine.}$$

GCD processors

Fig 5.5

HDL code for gcd

```
'''
gcd(in: x, Y ; out: z);
   register XR, YR, TEMPR;
    XR := x;                (Input data)
    YR : = Y;
    While XR>0 do begin
        if XR≤YR then begin
           TEMPR = YR;              (Swap YR and XR)
            YR = XR;
             XR = TEMPR; end
         XR:= XR - YR;          (Subtract ~~R~~ YR from XR)
      end
    z = YR;             (Output Result)
end gcd;
'''
```

Fig 5.6

Hardwire implementation of gcd.

Fig 5.7

State table of gcd processor

One-hot method

$S_0 = 0001$

$S_1 = 0010$

$S_2 = 0100$

$S_3 = 1000$

Control signals From the (Fig 5.7)

$D_0^+ = 0$

$D_1^+ = D_0 \cdot (XR > 0) \cdot \overline{(XR \geq YR)} + D_2 (XR > 0) \overline{(XR \geq YR)}$

$D_2^+ = D_0 (XR > 0) \cdot (XR \geq YR) + D_1 + D_2 (XR > 0) (XR \geq YR)$

$D_3^+ = D_0 \overline{(XR > 0)} + D_2 \overline{(XR > 0)} + D_3$

Subtract $= D_2$

Swap $= D_1$

Select XY $= D_0$

Load XR $= D_0 + D_1 + D_2$

Load YR $= D_0 + D_1$

Fig 5.10

ALL NAND one hot design.

Hardwired control

Design method

method 1:

- * Classical method of sequential circuit
- * Minimize the amount of hardwire in particular by using only $[\log_2 P]$ flipflops.

Method 2:

- * 1 flipflop per state
- * Known as one-hot-method
- * Expensive in terms of flipflops
- * Simplifies CU design and debugging.

State table

fig 5.4

* mealy machine
* moore machine

$\}$ → finite state machine.

1. Construct a P row state table that defines the desired input output behavior.

2. Select the minimum number of D flipflops and assign a p bit binary code to each state.

3. Design a combinational circuit c that generates the primary output signals $\{z_i\}$ and the secondary outputs $\{D_i\}$ that must be applied to the flipflops.

Fig 5.8) Excitation table

From excitation table get all the control signal.

Fig 5.9) ALL-NAND classical design of gcd processor

Organization:

A RAM is reffered to as a $2^m \times w$-bit or 2^m word memory

The RAM operates as follows:

First the ~~ta~~ address of the target location is transferred via address bus to the RAMs address buffer. The address is then processed by address decoder. Which selects the required location in the storage cell unit.

(N.B.: Every cell is connected to a set of data, address, ctrl signal.)

Fig 6.7) 1-D RAM unit.

Fig 6.8) 2-D RAM unit.

Semiconductor RAM:

Both bipolar ~~one~~ ~~f~~ and MOS technologies used ~~a~~ in RAMs but MOS was dominant in large RAM.

Two categories:

1. SRAM
2. DRAM

Fig 6.9) Static and dynamic RAM cells in MOS technologies.

Fig 6.10)- RAM IC Major external connection.

* Peak data-transfer rate

Fig 6.14 → Rambus DRAM interface.

Multilevel memories:

Semiconductor SRAM for cache
Semiconductor DRAM for main memory
Magnetic disk units for secondary memory.

Split cache < I cache for Instruction
D cache for Data

Fig 6.21 → Common memory hierarchy.

Virtual memory:

The term virtual memory is applied when the main and secondary memories appear to a user program like a single, large, and directly addressable memory.

Three reasons for using virtual memory:

1. Free from the need of carry out storage allocation permit efficient sharing of the available space.

2. Make programs independent of the configuration and capacity, allow seamless overflow.

3. To achieve very low access time and cost per bit.

1. CPU Registers
   ↳ Single clock cycle acccess
   ↳ Temporary storage for instruction and data

2. Main (Primary memory)
   ↳ Fast external memory
   ↳ Stores program or data that are in active use.
   ↳ Access time 5 or more clock cycle.

3. Secondary memory
   ↳ Acts as over-flow memory
   ↳ Very large in size

4. Cache
   ↳ Positioned between CPU and main memory.
   ↳ Access time one to three cycle
   ↳ Reside on the same IC as the CPU
   ↳ Essential for high performance computer.

Fig 6.1)

the stored

nondestructive readouts (NDRO)

Fig: 6.5 memory restoration of (DRO)

Memories that requires periodic refreshing are called dynamic memories.

- Static memories does not require refreshing.

A memory is volatile if the loss of power destroys the stored information.

DRAM ⇒ High access time, low cost
SRAM → Low access time, high cost

Dynamic memory refreshing = Data restore in DRO
Fig (6.5)

Fig 6.6) Characteristics of memory technologies.

$\log_2$(block/Line size)

Tag directory size = No. of lines × no. of tag bits.

No. of tag bits = $\log_2$(M.M size/cache size)

Communication method:

1. Intrasystem communication:
   - * Occurs within single computer system.
   - * Information transfer over distances less than a meter.
   - * Primarily implemented by groups of electrical wires called buses.
   - * Support parallel, word by word data transmission.

2. Intersystem communication:
   - * Involve communication over much longer distances.
   - * Electrical cable, fiber cable, wireless links.
   - * Serial (bit by bit) data transmission.

A set of computers and other system components that are linked together over relatively long distances constitute a computer Network.

Data " " : 16-128 bits

Fig 7.1 >> Communication within a computer via single shared bus.

Fig 7.2 >> System bus of powerPC 603.

The principle use of system bus is high speed data transfer between CPU and M.

Fig 7.3 >> Computers with separate system and IO busses

IO controller perform serial to parallel or parallel to serial conversion.

SCSI (Small Computer System Interface):

- * ANSI (American National Standard Institute) standard IO Bus.
- * Transfer data a byte at a time at rates up to 5 MB/s.

Fig 7.4 - SCSI IO bus.

Fig 6.39) - Table

* CPU generally has direct access to both M_1 and M_2. where as it doesnot have direct access to M_3.

Fig 6.40) - Basic structure of cache.

Fig 6.41) - look aside and look through organization.

Mapping:

Direct memory mapping:

P.A bits = $\log_2$(Number of word in M.M.)

Number of Blocks in M.M. = $\frac{\text{M.M. Size}}{\text{Block size}}$

In P.A bits

MSB $\log_2$(Number of blocks) bits ~~are~~ identifies the Block number.

Rest bits identifies the word in that block.

For cache,

Line size = Block size

No. of lines in cache = $\frac{\text{Cache size}}{\text{Line Size}}$

$$\text{Cost} = \frac{\text{Price}}{\text{Storage}} \text{ (dollers/bit)}$$

The performance of an individual memory device is primarily determined by the rate at which informatio can be read from or written into the memory.

Fig 6.2

Memory retention :

Memories whose content cannot be altered on line are read only memory (ROM) (Non erasable storge device.

Memories whose content can be changed off line with some difficulty are called programmable read only memory. (PROM)

In some memory the method of reading the memory destroy the stored information; this phenomenon is called destructive readout (DRO

, while $N' > N$ and/on $w' > w$.

↳ Constnuct $p \times q$ anray of $M_{N,w}$ ICs, where $p = \lceil N'/N \rceil$ and $q = \lceil w'/w \rceil$

Fig 6.11) → Incneasing the wond size of a RAM by a facton of q.

Fig 6.12) → " " numben of wond " " " "

Fig 6.13) → Commencial 8M × 8bit DRAM Chip

Fast RAM intenfaces:

Two basic ways to incnease the data tnansfen nate acnoss its extennal intenface by a facton of S.

1. Use a biggen memony wond,
   $w = Sn$ bits can be accessed in one memony cycle T_m.

2. Access mone than one wond at a time.
   we pantition the RAM into S separate banks $M_0, M_1, - M_{S-1}$.

The RAMBUS DRAM and Intenface:

- ✱ Tnansfen memony data at high speed oven nannow processon-memony link.
- ✱ Speedup technique → Syncnonous intenface, addness intenleaving and caching.

as a local or expansion bus.

* PCI (Peri^{ph}feral Component Interconnect) which can transmit 4 or 8 bytes words at rates of 500 MB/s or more.

Bus Interfacing:

Buffer circuits called bus drivers an recievers are needed to transfer signals to and from the bus respectively.

Tristate buffer (0,1, z(high impedence)) serves as a bus line drivers. (Fig 7.13)

Advantages of tristate logic circuit:

- Greatly increase the fan In, fan Out limits, permitting large number of devices to be attached on the same line.
- Support bidirectional transmission over the bus by allowing the same bus connection to serve as an input port and as an output port at different times.