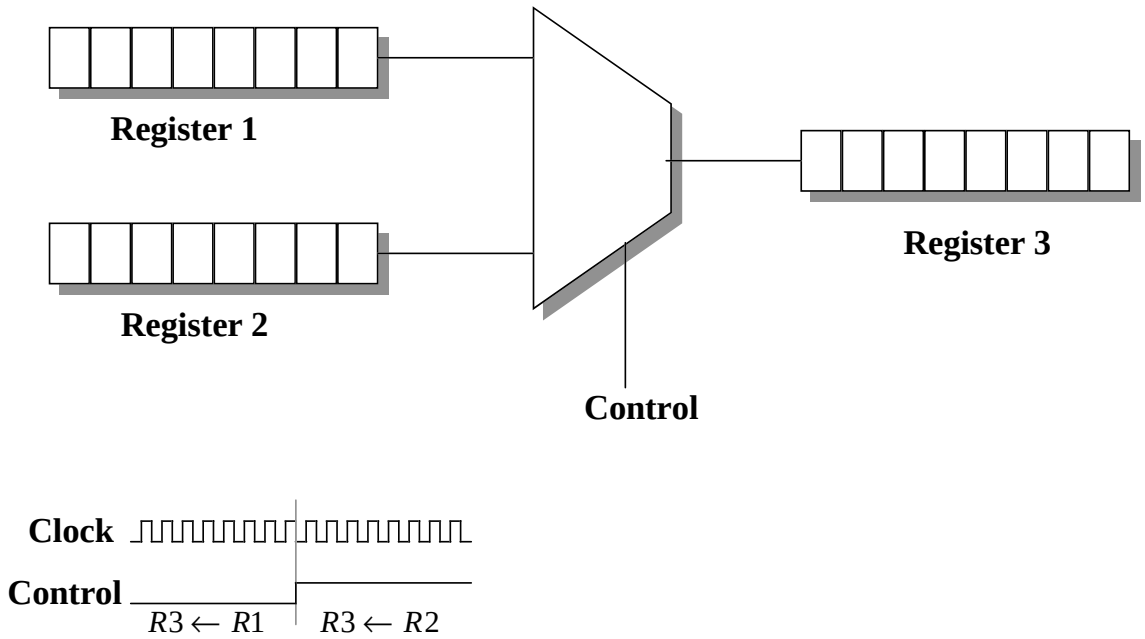
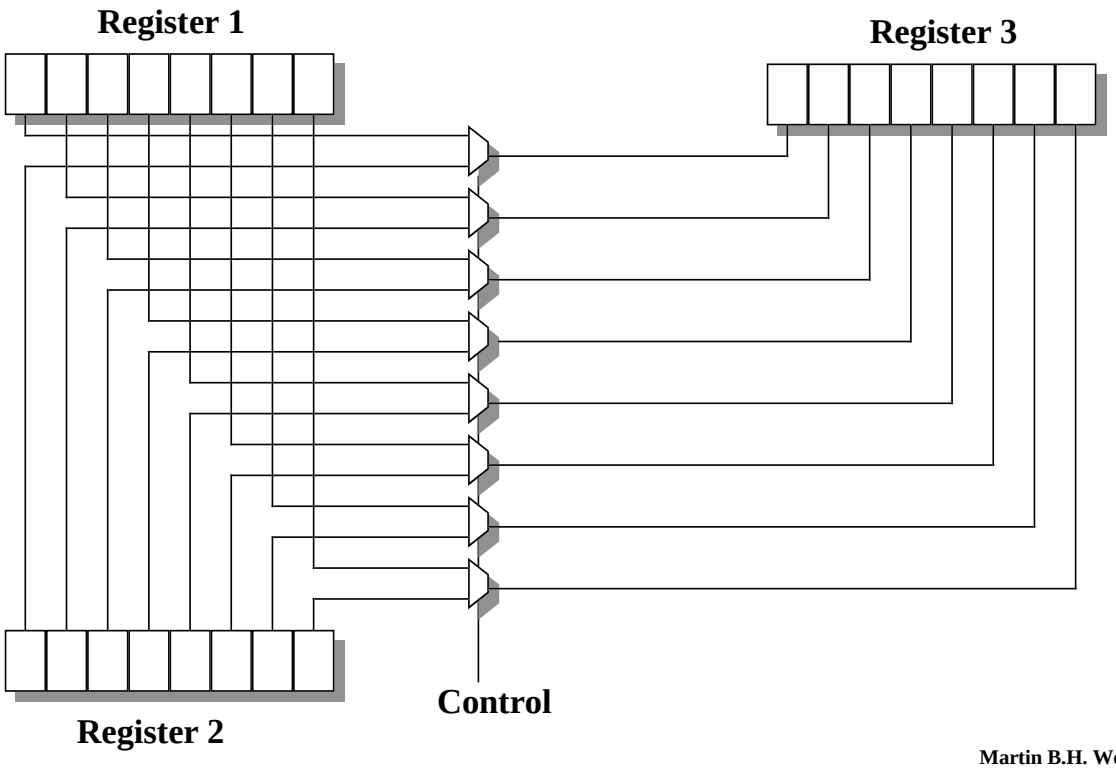
Transferring Data Among Registers

- **Point-to-Point Transfer**
  - **Serial Transfer: Shift Registers**
  - **Parallel Transfer**
    - **Control (Strobing Signal)**
    - **Clocking**
  - **Multiplexing**
- **Bus Transfer**
  - **Avoids Using Multiplexers**
  - **Destination Device is Selected**
  - **Registers are Interconnected With a Common Bus**

Serial Data Register Transfer

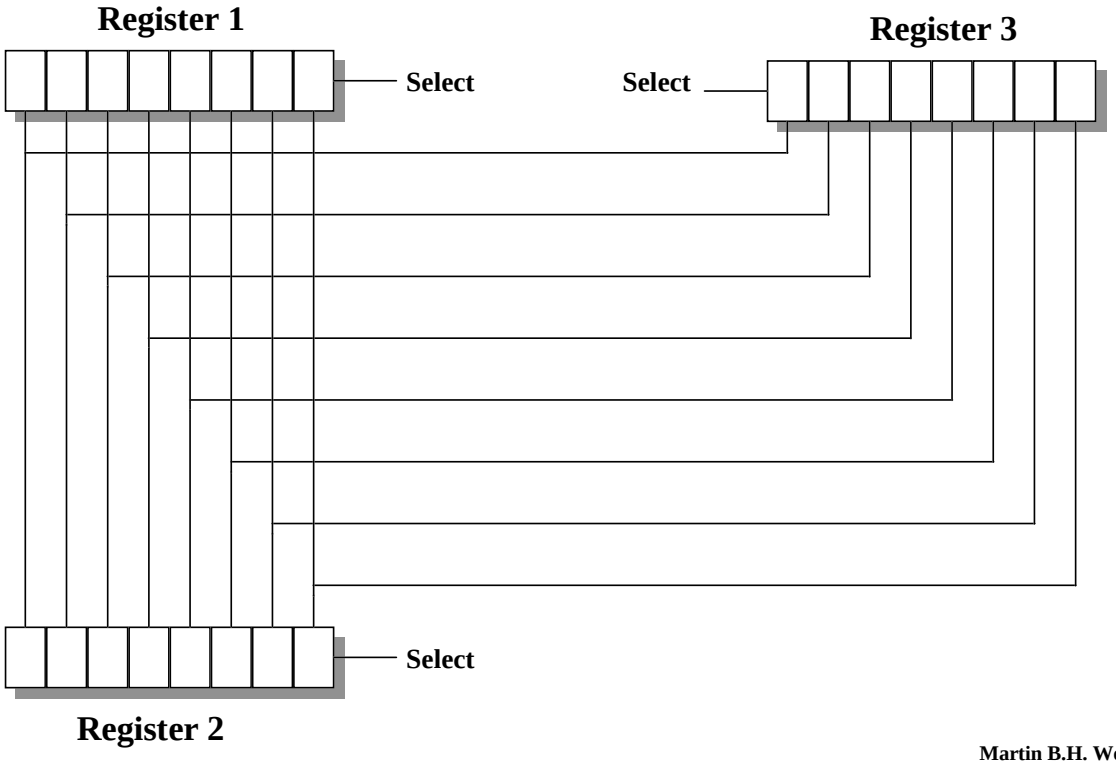


Parallel Data Transfer



Martin B.H. Weiss

Parallel Data Transfer



Martin B.H. Weiss

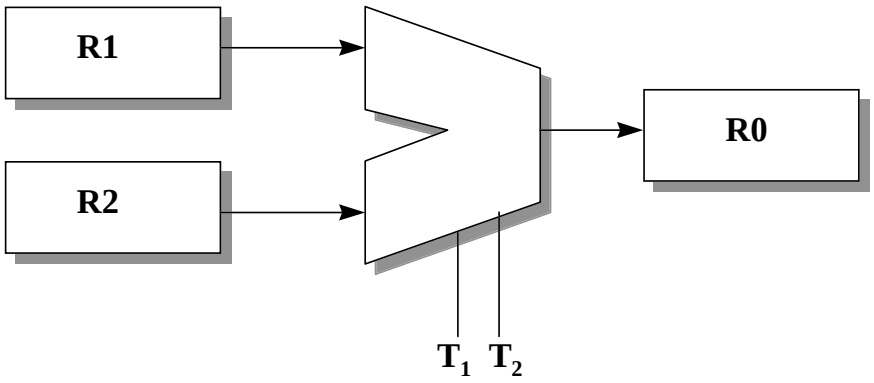
Register Transfer Definitions

- A Register Is A Group of Flip-Flops Capable of Storing Information and Performing Simple Operations
- Micro-operation
  - Elementary Operation
  - Performed in Parallel on Data Stored in a Register
  - Performed in One Clock Period
  - Examples
    - Load
    - Shift Right
    - Shift Left

Register Transfer Notation

- Symbolic Method for Representing Information Flow Among Registers
- Examples
  - Register 2 Gets the Contents of Register 1: R2← R1
  - Register 2 Gets Part of Register 1: R2(0-7)←R1(8-15)
  - Register 1 Gets Data at Memory Location AR: R2← M[AR]

Hardware Structure



Transfer Notation

- Need Multiplexer Control Leads
- Condition Register Transfer by State of the Control Leads
- Notation Examples
  - T_1T_2: R0←R1
  - $T_1'T_2$: R0←R1
- External Control Logic Determines the Values of the Leads
- Need $\log_2 N$ Control Leads for N Multiplexers

Application of Register Transfer

- Formalization of Logic
- Translation of Hardware Design into Useful Operational Structures
- Stages of Representation
  - Hardware Schematic
    - Can Produce Optimal Design
    - Difficult to Comprehend Operational Aspects
  - Register Transfer Notation
    - Simple Operational Formalization of Hardware Functionality
    - Needs Extension to be Useful
  - Microoperations
    - Extension of RTN
    - Building Blocks for ALU and CPU Functionality

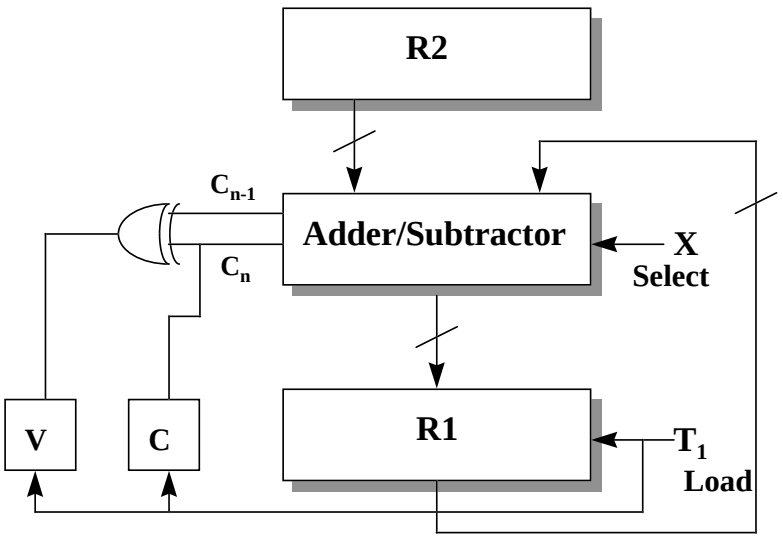
Microoperations

- Definition: Elementary Operation Performed on Registers in a Single Clock Cycle
- Classes
  - Register Transfer
  - Arithmetic
  - Logic
  - Register Shift

Arithmetic Operations

- R0 ← R1 + R2 (Addition)
- R0 ← R1 - R2 (Subtraction)
- R2 ← R2' (One's Complement)
- R2 ← R2' +1 (Two's Complement)
- R1 ← R1 + 1 (Increment)
- R1 ← R1 - 1 (Decrement)

Arithmetic Hardware Structure



Logic Operations

- R0 ← R0' (Complementation)
- R0 ← R1 ∧ R2 (AND)
- R0 ← R1 ∨ R2 (OR)
- R0 ← R1 ⊕ R2 (XOR)

Shift Operations

- R ← shl R
- R ← shr R
- R ← rol R
- R ← ror R
- R ← asl R
- R ← asr R
- Discussion
  - Shift: End Receives the Input From the Serial Input
  - Rotate: End Receives Shifted-Out Information (No Loss of Bits)

Arithmetic Shift

- Left - Multiply a Signed Binary Number by 2
- Right - Divide a Signed Binary Number by 2
- Leave the Sign Bit Unchanged (R_{n-1})
- Negative Numbers
  - In Two's Complement Form
  - Sign Bit = 1
  - Insert a 0 Into the Vacated Bit

Bus Transfer

- Bus
  - Common Set of Parallel Wires (One for Each Bit) For All System Components
  - Also Contains Control Wires
- Operation
  - Device Outputs on the Bus Can Have Three States (1,0, High Impedance)
  - If the Device is Selected, the Output Must Be a 1 or a 0
  - If Not, the Output is High Impedance
  - Selection Done Via a Decoder

Memory Transfer

- Transfer Data From a Memory Location to a Register
- Address
  - Address Bus
  - Address Register (AR)
  - Bus Buffer
- Memory
- Data
  - Data Bus
  - Bus Buffer
  - Registers