Register Transfer

Types of Register

- A **Register** is a group of flip-flops with each flip-flop capable of storing **one bit** of information. An *n-bit* register has a group of *n flip-flops* and is capable of storing binary information of *n-bits*..
- Registers are a type of computer memory used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU.
- The registers used by the CPU are often termed as Processor registers.
- A processor register may hold an instruction, a storage address, or any data (such as bit sequence or individual characters).
- The computer needs processor registers for manipulating data and a register for holding a memory address.

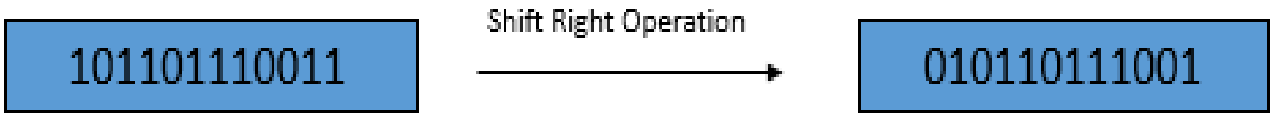
Register	Symbol	Number of bits	Function
Data register	DR	16	Holds memory operand
Address register	AR	12	Holds address for the memory
Accumulator	AC	16	Processor register
Instruction register	IR	16	Holds instruction code
Program counter	PC	12	Holds address of the instruction
Temporary register	TR	16	Holds temporary data
Input register	INPR	8	Carries input character
Output register	OUTR	8	Carries output character

- The Data Register (DR) contains 16 bits which hold the operand read from the memory location.
- The Memory Address Register (MAR) contains 12 bits which hold the address for the memory location.
- The Program Counter (PC) also contains 12 bits which hold the address of the next instruction to be read from memory after the current instruction is executed.
- The Accumulator (AC) register is a general purpose processing register.
- The instruction read from memory is placed in the Instruction register (IR).
- The Temporary Register (TR) is used for holding the temporary data during the processing.
- The Input Registers (IR) holds the input characters given by the user.
- The Output Registers (OR) holds the output after processing the input data.

Register Transfer Language (RTL)

➢Register Transfer Language (RTL) : The **Register Transfer Language** is the symbolic representation of notations used to specify the sequence of micro-operations.

- The result of the operation may be:
  - replace the previous binary information of a register or.
  - transferred to another register.



Register Transfer

- Information transfer from one register to another is described by a *replacement operator:*

  R2 ← R1 (← acts as a replacement operator)

- The statement R2← R1 denotes a transfer of the content of register R1 into register R2.
- It designates a replacement of the content of R2 by the content of R1.
- The content of the R1 (source) does not change after the transfer
- If we want the transfer to occur only under a predetermined control condition then it can be shown by an if-then statement.

  if (P=1) then R2← R1

Register Transfer cont

- P is the control signal or Control Function generated by a control section.
- Control function is a Boolean variable that is equal to 0 or 1.
- Control function is included in the statement as

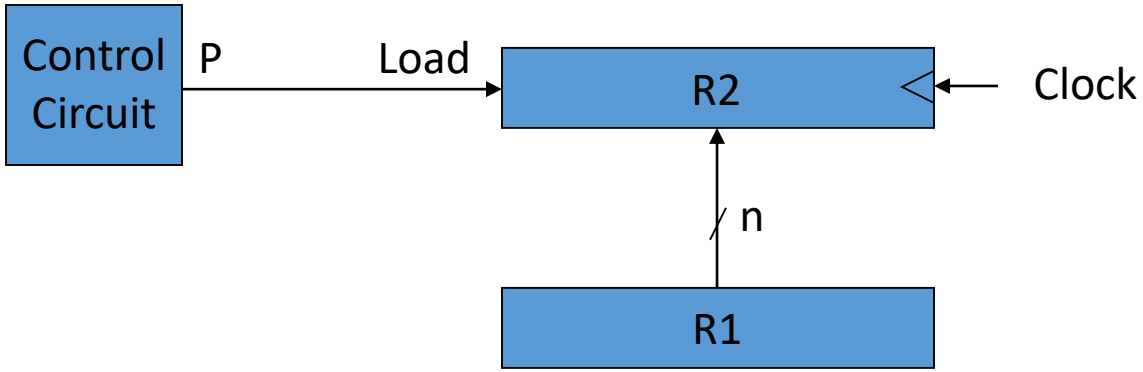
P: R2← R1

- Control condition is terminated by a colon implies transfer operation be executed by the hardware only if P=1.
- Every statement written in a register transfer notation implies a hardware construction for implementing the transfer.

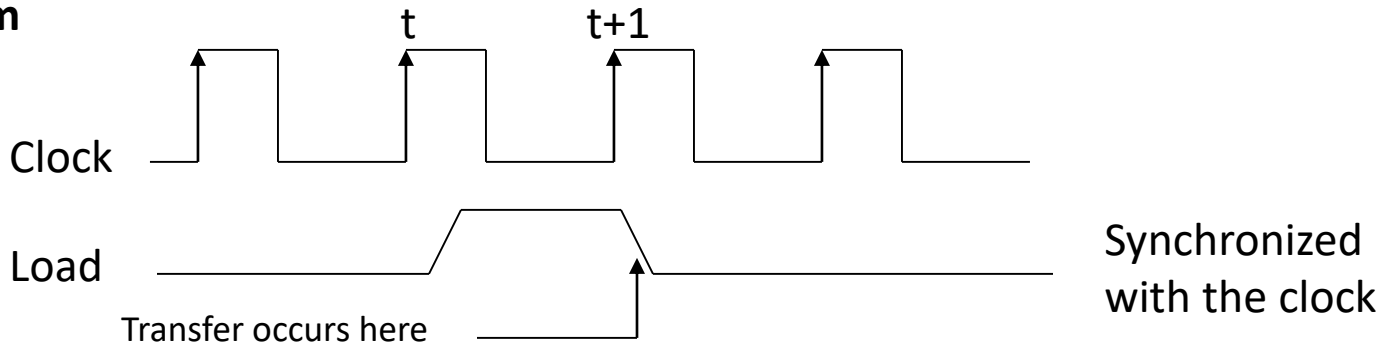
Register Transfer cont.

Hardware implementation of a controlled transfer: P: R2 ← R1

Block diagram:



Timing diagram



Register Transfer cont.

- The n outputs of register R1 are connected to the n inputs of register R2.
- The letter n will be used to indicate any number of bits for the register. It will be replaced by an actual number when the length of the register is known.
- Register R2 has a load input that is activated by the control variable P.
- It is assumed that the control variable is synchronized with the same clock as the one applied to the register.
- As shown in the timing diagram, P is activated in the control section by the rising edge of a clock pulse at time *t*.
- The next positive transition of the clock at time *t* + 1 finds the load input active and the data inputs of R2 are then loaded into the register in parallel.
- P may go back to 0 at time *t*+1; otherwise, the transfer will occur with every clock pulse transition while P remains active.

Transfer cont.

➢**Register Transfer Operations:**

- The operation performed on the data stored in the registers are referred to as register transfer operations.
- There are different types of register transfer operations:
- **1. Simple Transfer – R2 <- R1**
- The content of R1 are copied into R2 without affecting the content of R1. It is an unconditional type of transfer operation.

2. Conditional Transfer –

- Representation of a (conditional) transfer

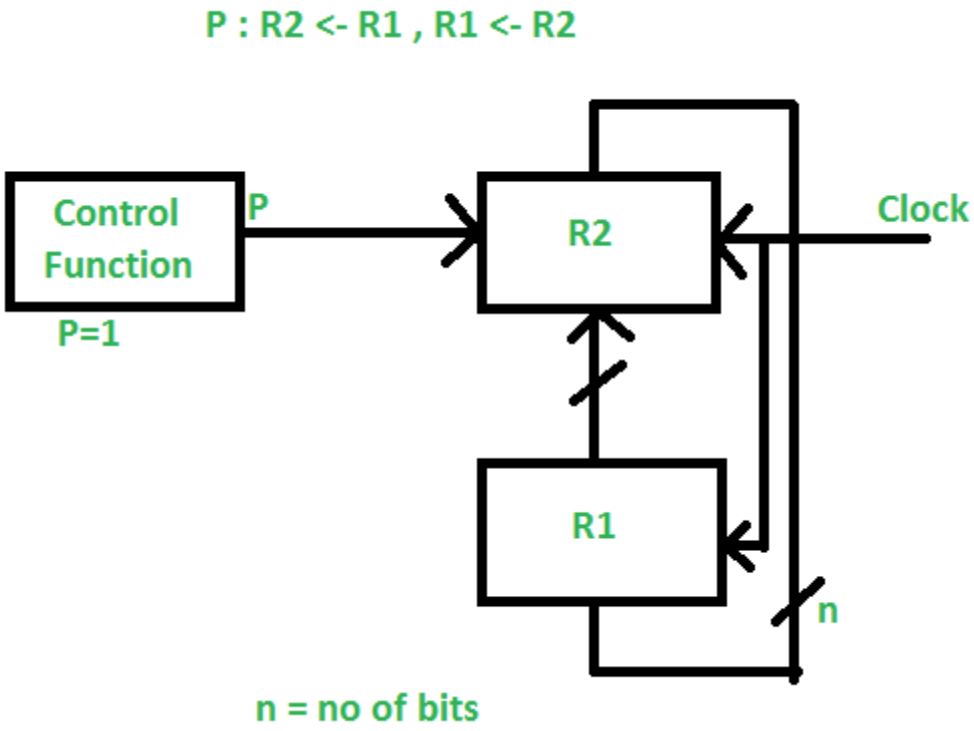
  Control Function P: R2 ← R1

- A binary condition or control function (P equals to 0 or 1) determines when the transfer occurs.
- The content of R1 is transferred into R2 only if P is 1.
- Every statement written in a register transfer notation implies a H/W construction for implementing the transfer.

Register Transfer cont.

3. Simultaneous Operations –

- If 2 or more operations are to occur simultaneously then they are separated with comma **(,)**.



If the control function P=1, then load the content of R1 into R2 and at the same clock load the content of R2 into R1.

SIMULTANEOUS OPERATIONS

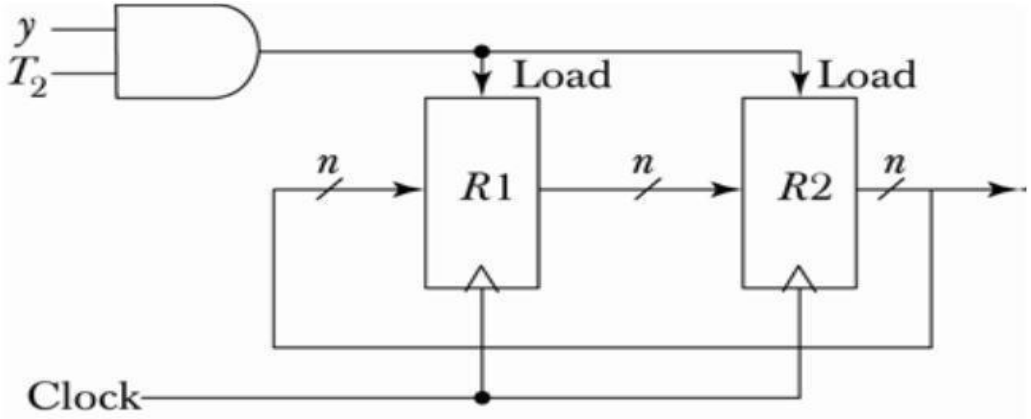
- If two or more operations are to occur simultaneously, they are separated with commas

P: $R3 \leftarrow R5$, $MAR \leftarrow IR$

- Here, if the control function P = 1, load the contents of R5 into R3, and at the same time (clock), load the contents of register IR into register MAR.

Questions

- Show the block diagram of the hardware that implements the following register transfer
- statements. yT2: R2 ← R1, R1 ← R2



- Consider the following register transfer statements

xT: R1<-R1 + R2 (if x = 1); xT: R1 <- R2 (if x = 0)

where T represents clock. Draw a diagram showing the hardware implementation of the above statements