Final Register Set Design

Nahin Ul Sadad
Lecturer
CSE, RUET

Register Set in CPU

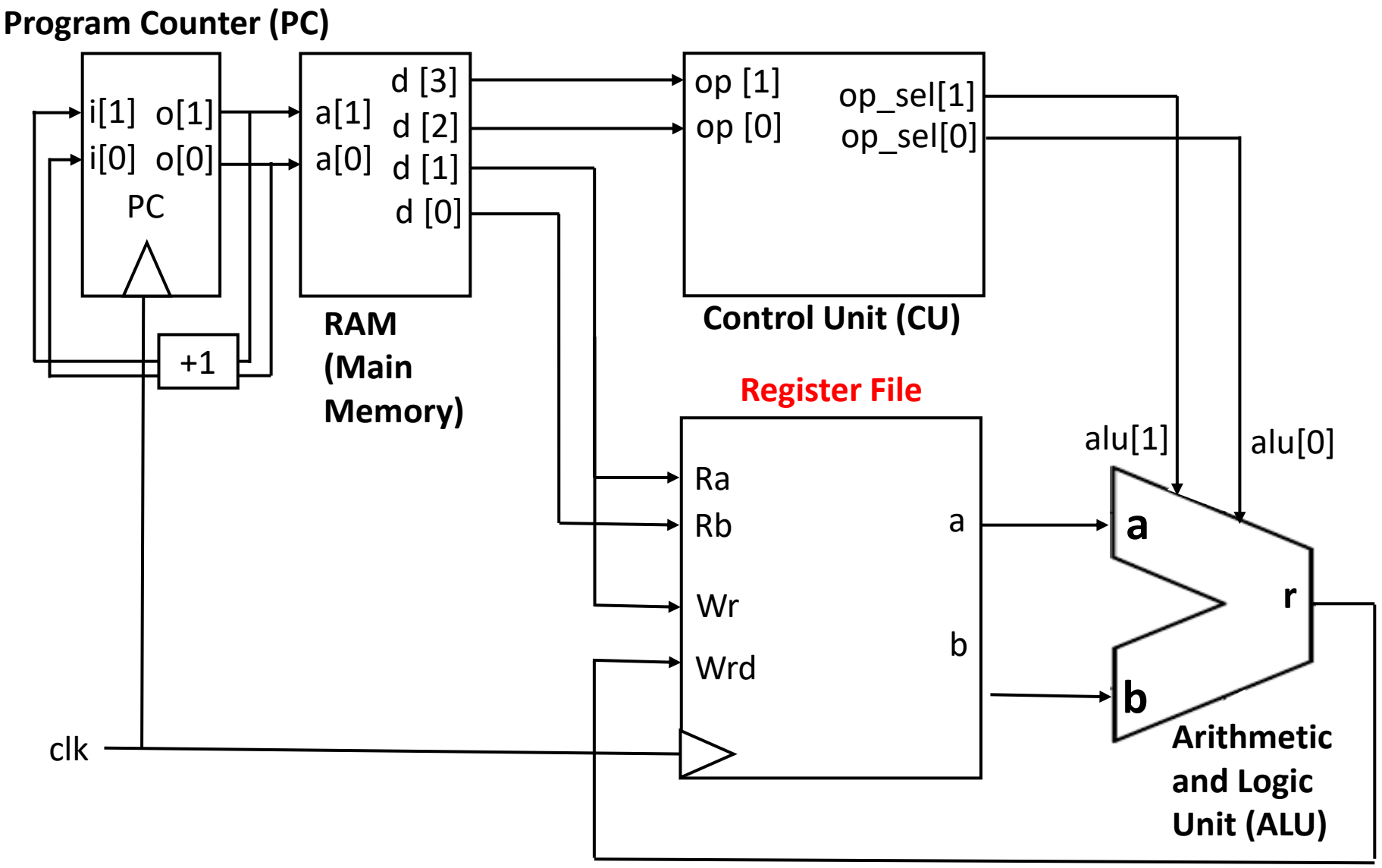


Figure: 1-bit CPU

1. Program Counter will have address of next instruction to be executed in current clock cycle.
2. Address in PC will be sent to RAM to retrieve instruction.
3. Instruction will be decoded by control unit and will select **registers** and/or immediate values.
4. Data within **registers** and/or immediate values will be sent to Arithmetic and Logic Unit (ALU) to perform operations.
5. ALU will perform operation and result will be sent to the **register** to be written.
6. Finally, PC will be incremented to point to the next instruction in next clock cycle.

Prerequisites for Register Design

D Flip-flop

D	Q(t + 1)
0	0
0	1

Fig: D Flip-flop truth table

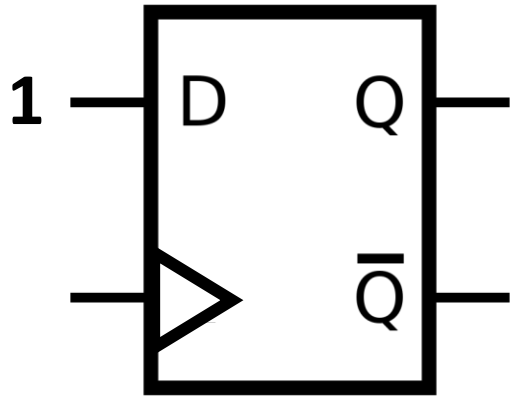


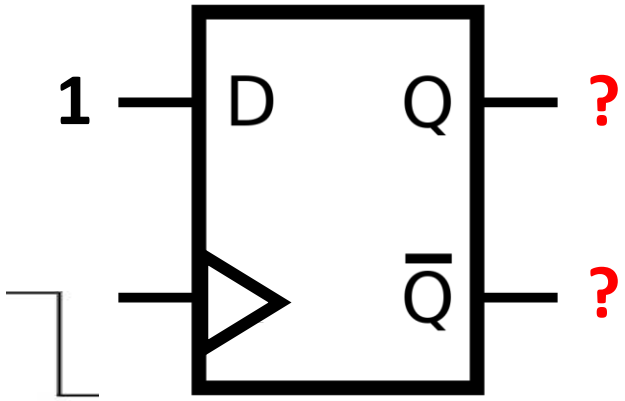
Fig: D Flip-Flop

D Flip-Flop

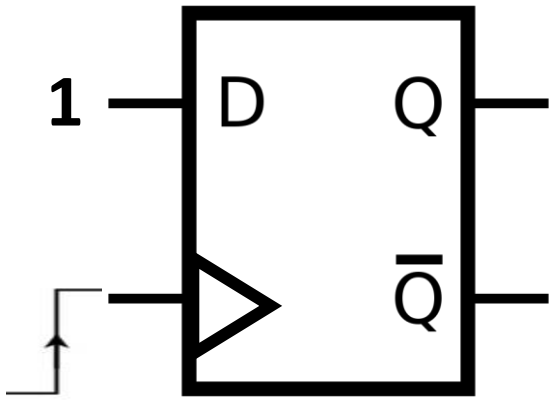
Assume, Q=1 which means D-FF has 1 value stored.

D	Q(t + 1)
0	0
0	1

Fig: D Flip-flop truth table



At Negative Clock Edge



At Positive Clock Edge

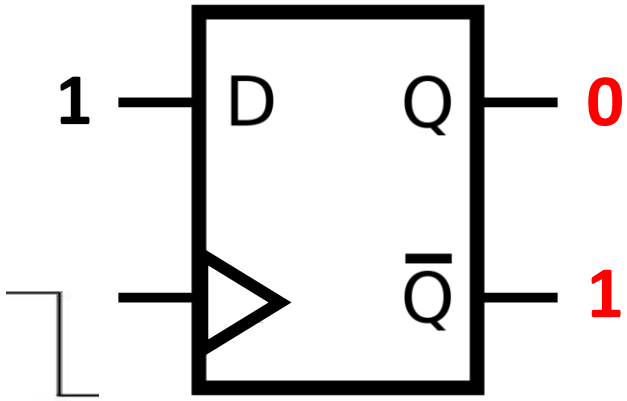
Fig: How D Flip-Flop works.

D Flip-Flop

Assume, Q=1 which means D-FF has 1 value stored.

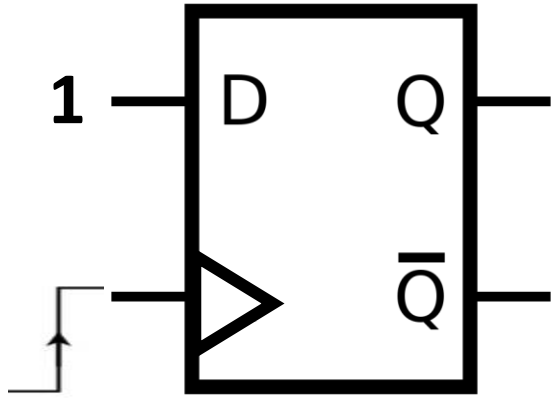
D	Q(t + 1)
0	0
0	1

Fig: D Flip-flop truth table



At Negative Clock Edge

D Flip-flop will not update its content because clock is not on positive edge.



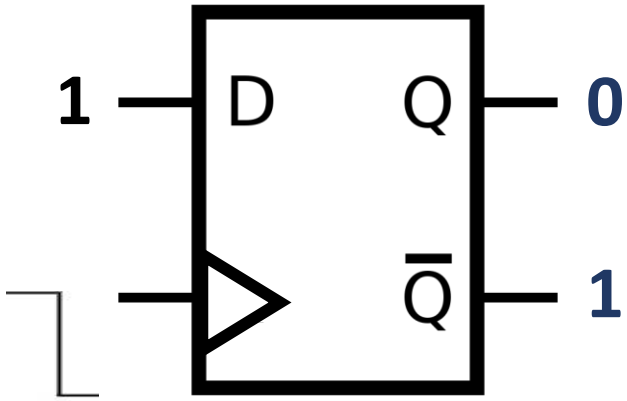
At Positive Clock Edge

Fig: How D Flip-Flop works.

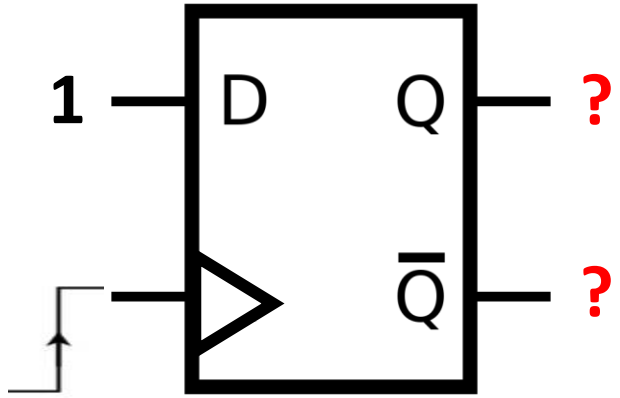
D Flip-Flop

D	Q(t + 1)
0	0
0	1

Fig: D Flip-flop truth table



At Negative Clock Edge



At Positive Clock Edge

Fig: How D Flip-Flop works.

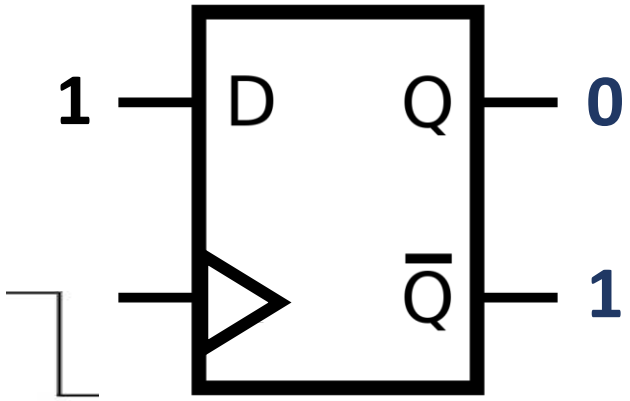
Assume, Q=1 which means D-FF has 1 value stored.

D Flip-flop will not update its content because clock is not on positive edge.

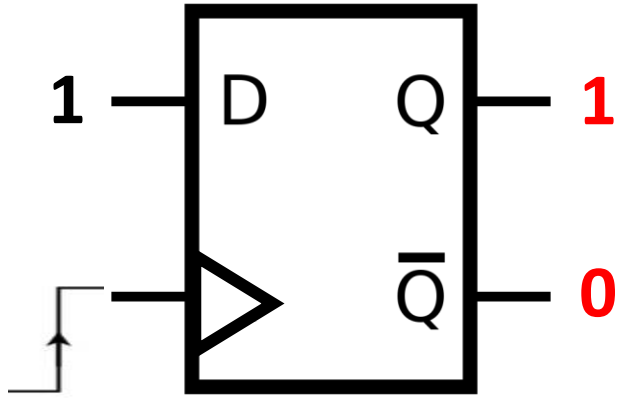
D Flip-Flop

D	Q(t + 1)
0	0
0	1

Fig: D Flip-flop truth table



At Negative Clock Edge



At Positive Clock Edge

Fig: How D Flip-Flop works.

Assume, Q=1 which means D-FF has 1 value stored.

D Flip-flop will not update its content because clock is not on positive edge.

D Flip-flop will update its content because clock is on positive edge.

Register Basics

Register

Register is a very fast computer memory, used to store data/instruction in-execution.

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.

There are several types of registers:

1. General Purpose Registers
2. Data Registers
3. Address Registers
4. Condition Codes/FLAG Registers

Register

There are several types of registers:

1. **General Purpose Registers:** General-purpose registers can be assigned to a variety of functions by the programmer.
2. **Data Registers:** Data registers can be used only to hold data and cannot be employed in the calculation of an operand address.
3. **Address Registers:** Address registers can be general purpose or can be devoted to a particular addressing mode.
    a. **Segment Pointer Registers:** In a machine with segmented addressing, a segment register holds the address of the base of the segment. There may be multiple registers: one for the operating system and one for the current process.
    b. **Index Registers:** Index registers are used for indexed addressing and may be auto-indexed.
    c. **Stack Pointer Register:** If there is user-visible stack addressing, then typically there is a dedicated register that points to the top of the stack. This allows implicit addressing which means that push, pop, and other stack instructions need not contain an explicit stack operand.

Register

4. **Condition Codes/FLAG Registers:** Condition codes/FLAG register which is at least partially visible to the user holds condition codes (also referred to as flags). Condition codes are bits set by the processor hardware as the result of operations.

   In addition to the result itself being stored in a register or memory, a condition code is also set. The code may subsequently be tested as part of a conditional branch operation.

   For example, an arithmetic operation may produce a positive, negative, zero, or overflow result.

   Common fields or flags include the following:

   a. **Sign:** It contains the sign bit of the result of the last arithmetic operation.

   b. **Zero:** It sets when the result is 0.

   c. **Carry:** It sets if an operation resulted in a carry (addition) into or borrow (subtraction) out of a high-order bit. Used for multiword arithmetic operations.

   d. **Equal:** It sets if a logical compare result is equality.

Register

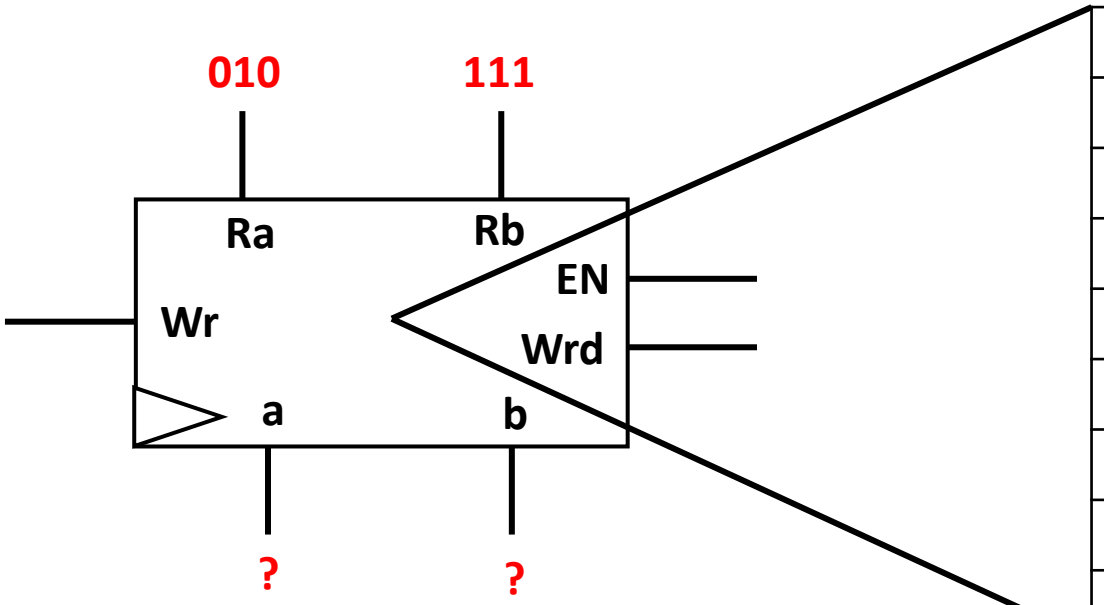
e. **Overflow:** It is used to indicate arithmetic overflow.

f. **Interrupt Enable/Disable:** It is used to enable or disable interrupts.

g. **Supervisor:** It indicates whether the processor is executing in supervisor or user mode. Certain privileged instructions can be executed only in supervisor mode and certain areas of memory can be accessed only in supervisor mode.

Register Set Review

4-bit Register Set (Reading)



Address	Register	Value stored
000	R0	0111 (7)
001	R1	0100 (4)
010	R2	0011 (3)
011	R3	0111 (7)
100	R4	1000 (8)
101	R5	0001 (1)
110	R6	1000 (8)
111	R7	0010 (2)

Suppose, Register set has eight 4-bit registers.

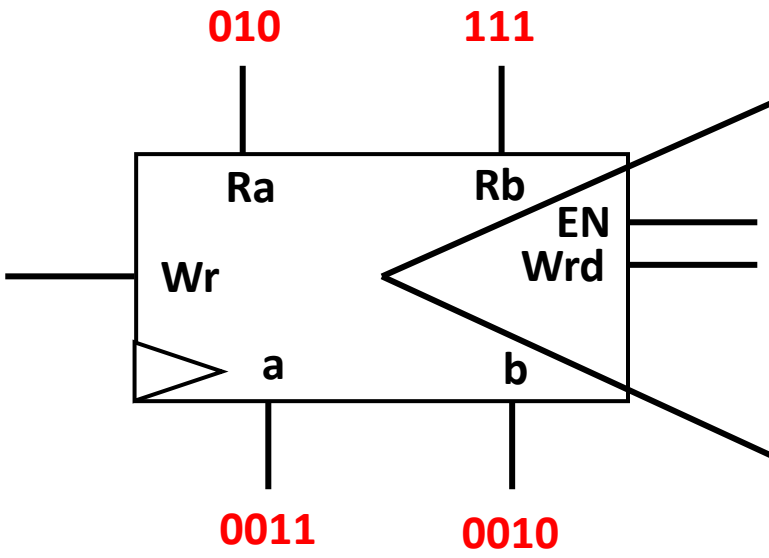
Since Register Set has 8 registers, it will need $log_2 8 = 3$ address lines.

Since registers are 4-bit registers, so it will store 4-bit value.

Reading in Register Set

Ra will take address of register and show value stored in that register in a (4-bit value).
and
Rb will take address of register and show value stored in that register in b (4-bit value).

4-bit Register Set (Reading)



Address	Register	Value stored
000	R0	0111 (7)
001	R1	0100 (4)
010	R2	0011 (3)
011	R3	0111 (7)
100	R4	1000 (8)
101	R5	0001 (1)
110	R6	1000 (8)
111	R7	0010 (2)

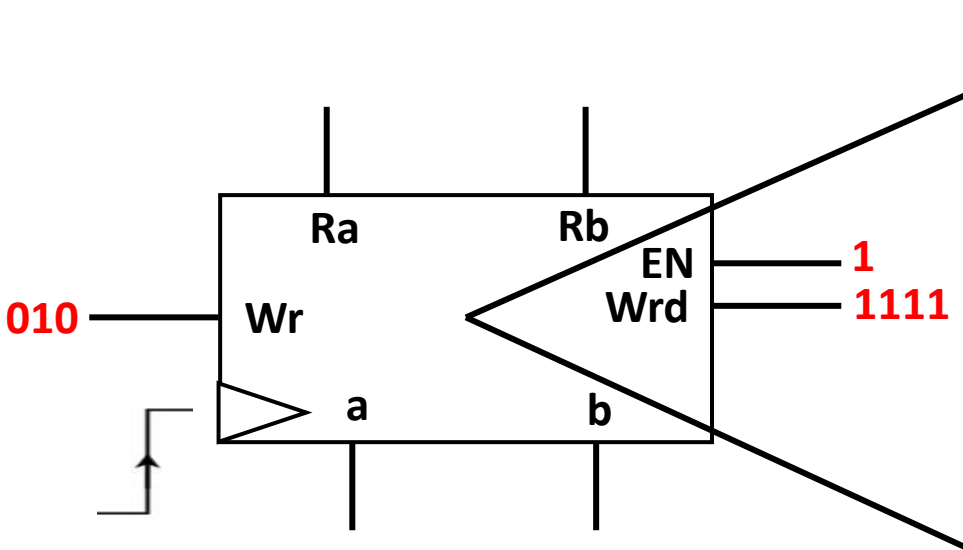
R2 register has 0011 (3) value stored.

R7 register has 0010 (2) value stored.

Reading in Register Set

Ra will take address of register and show value stored in that register in a (4-bit value).
and
Rb will take address of register and show value stored in that register in b (4-bit value).

4-bit Register Set (Writing)



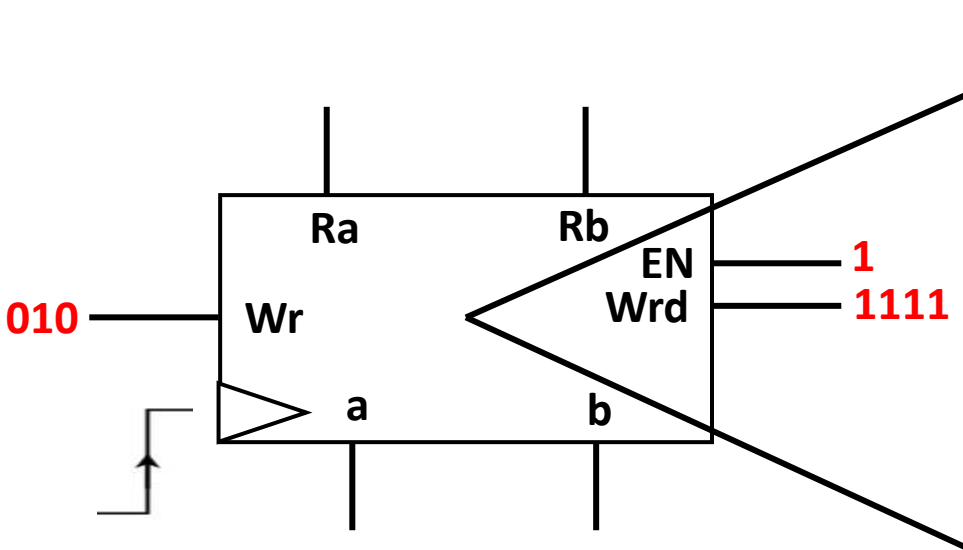
Address	Register	Value stored
000	R0	0111 (7)
001	R1	0100 (4)
010	R2	0011 (3)
011	R3	0111 (7)
100	R4	1000 (8)
101	R5	0001 (1)
110	R6	1000 (8)
111	R7	0010 (2)

What will happen?

Writing in Register Set

EN will enable/disable writing operation in register set. (1-Enable/0-Disable)
and
Wr will take address of register to be written.
and
Wrd will take (4-bit value) value to be written in Wr register.

4-bit Register Set (Writing)



Address	Register	Value stored
000	R0	0111 (7)
001	R1	0100 (4)
010	R2	1111 (15)
011	R3	0111 (7)
100	R4	1000 (8)
101	R5	0001 (1)
110	R6	1000 (8)
111	R7	0010 (2)

What will happen?

Writing in Register Set

EN will enable/disable writing operation in register set. (1-Enable/0-Disable)
and
Wr will take address of register to be written.
and
Wrd will take (4-bit value) value to be written in Wr register.

Final Register Set Design

Register Set

Register Set chip is shown below:

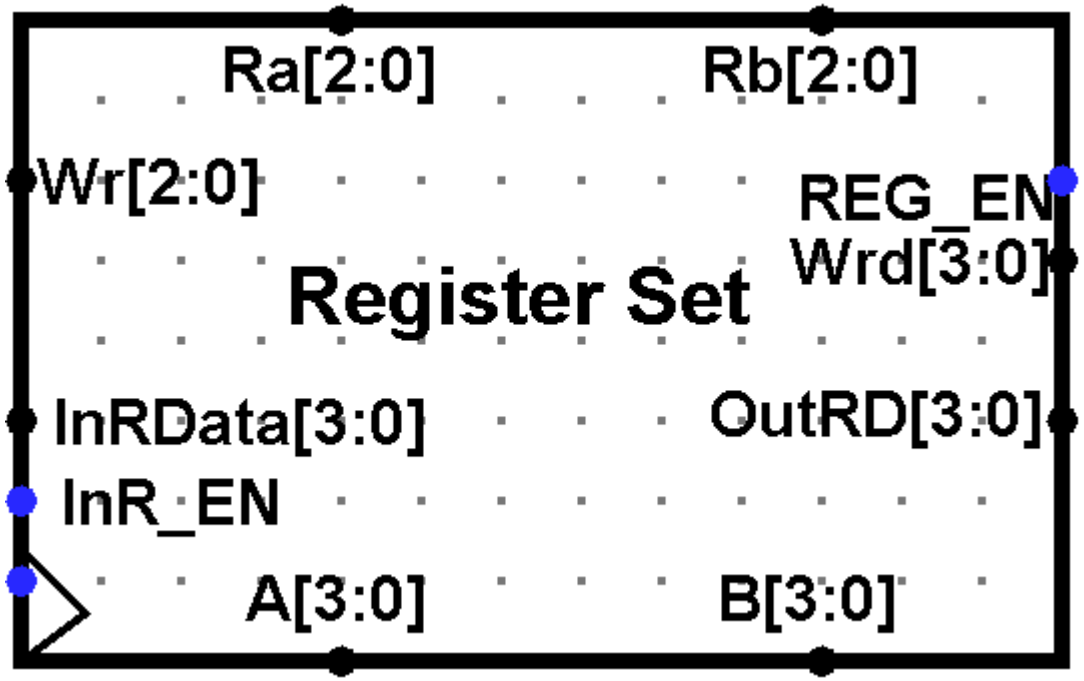


Figure: 4-bit Register Set with 8 Registers Chip

Here,
Ra = Register A
Rb = Register B
A = Data of Register A
B = Data of Register B

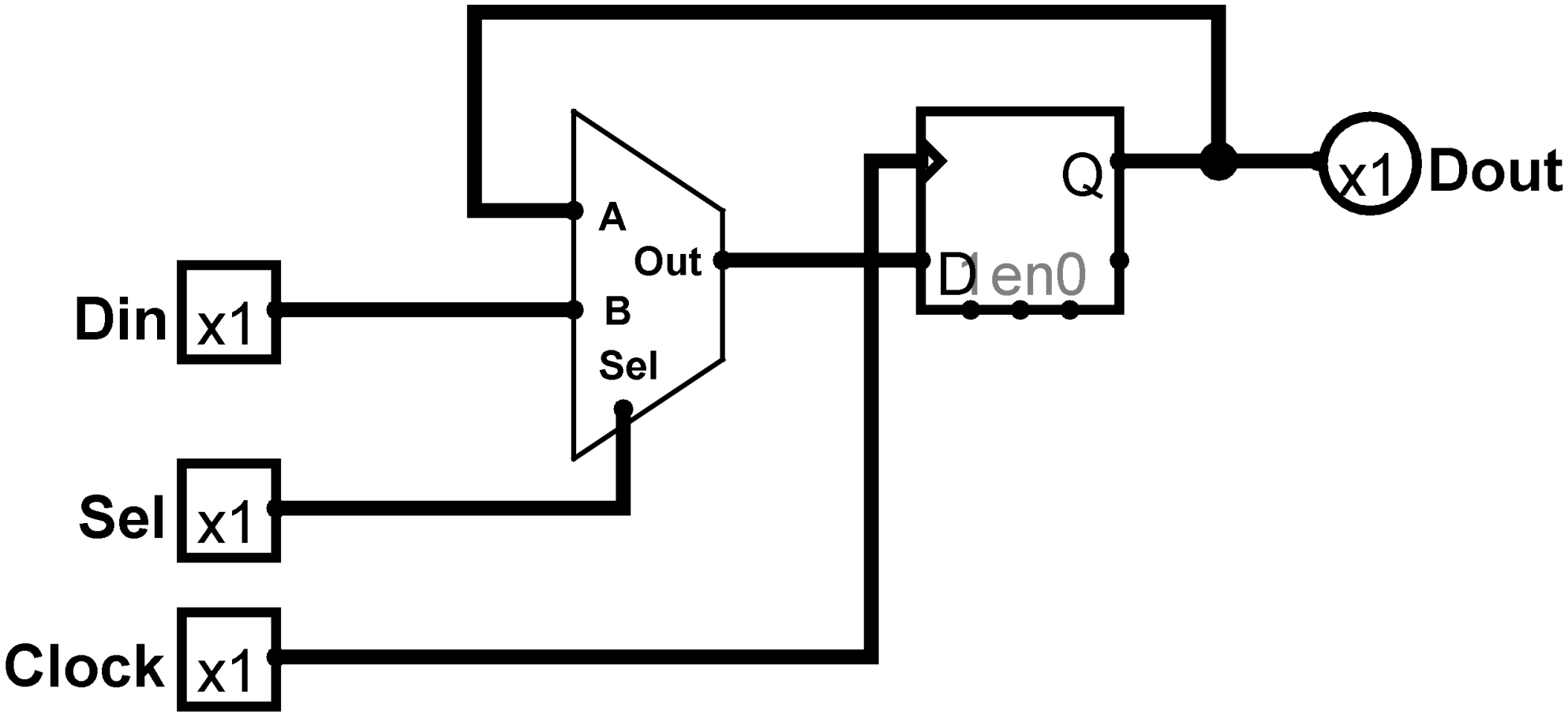
Wr = Register to be written
Wrd = Data to be written in Wr
REG_EN = Write in Register Enable

InRData = Data to be written in Input Register
InR_EN = Write Enable in Input Register

OutRD = Data in Output Register

1-bit Register Cell

1-bit Register



1-bit Register

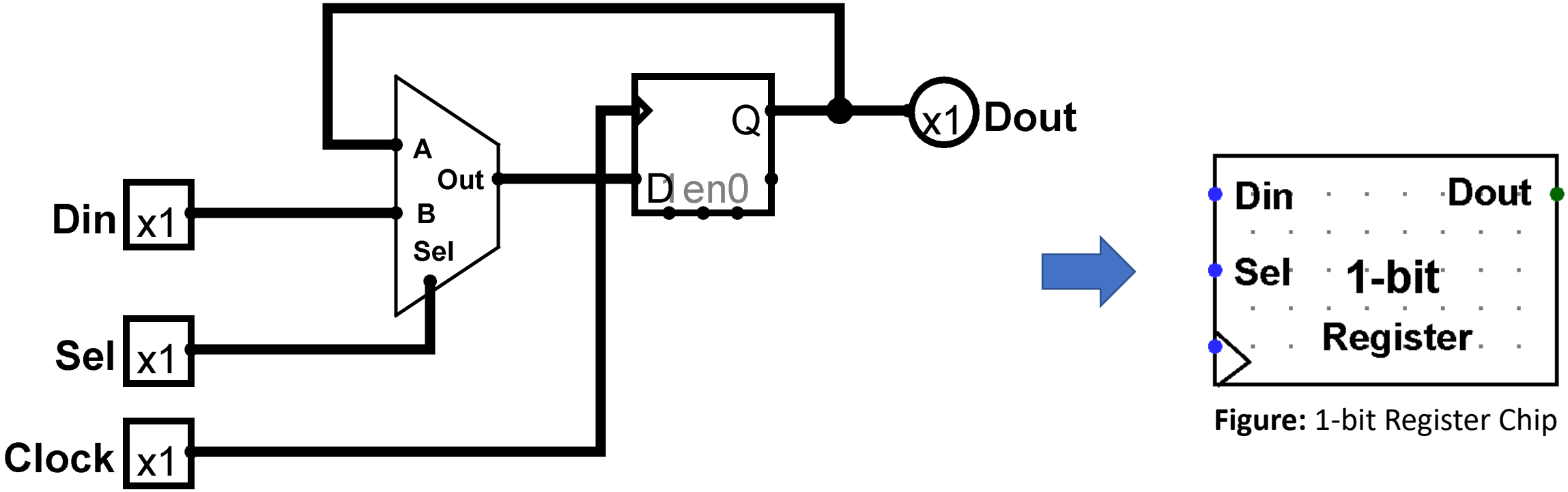
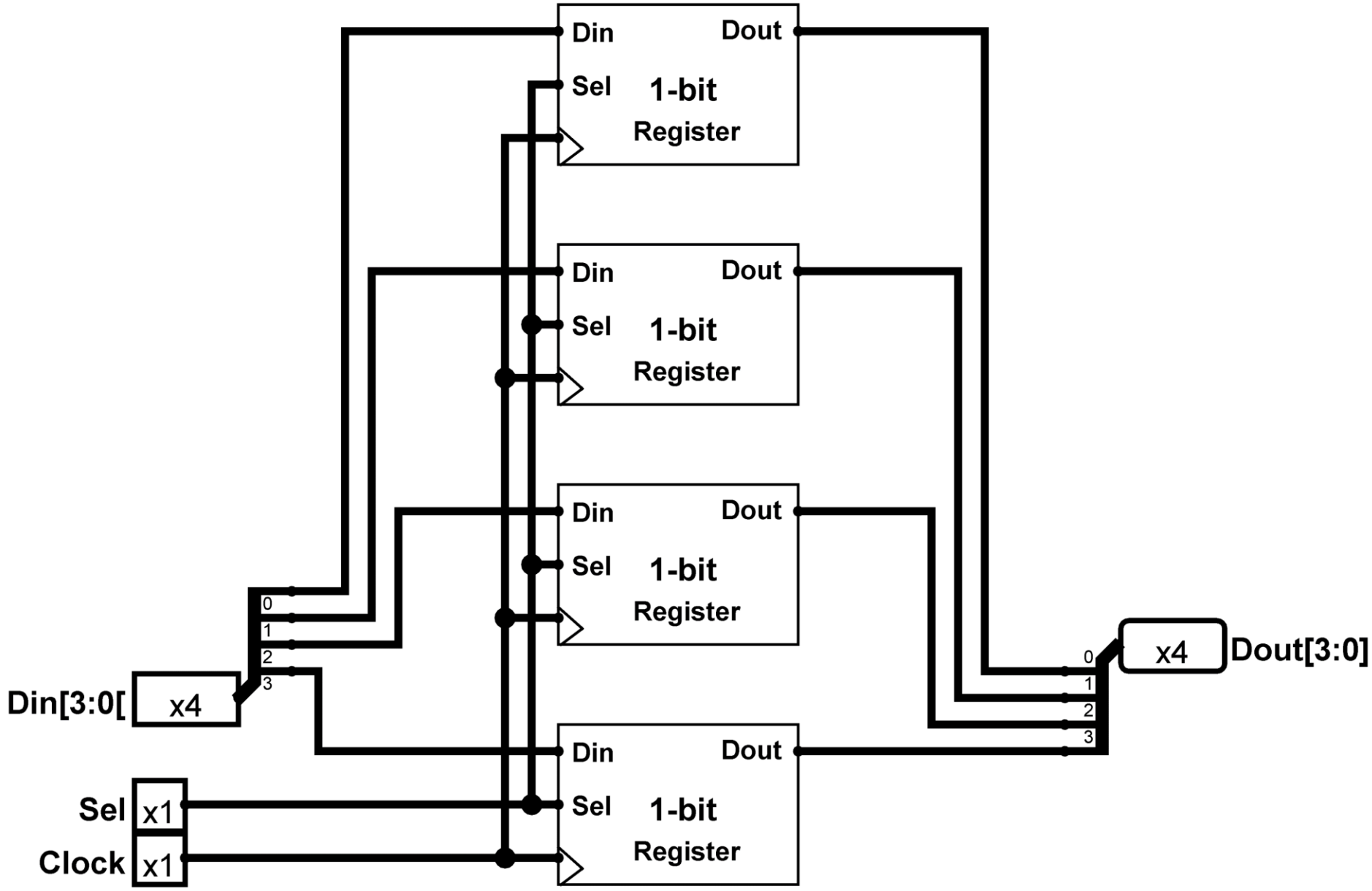


Figure: 1-bit Register Chip

4-bit Register Cell

4-bit Register



4-bit Register

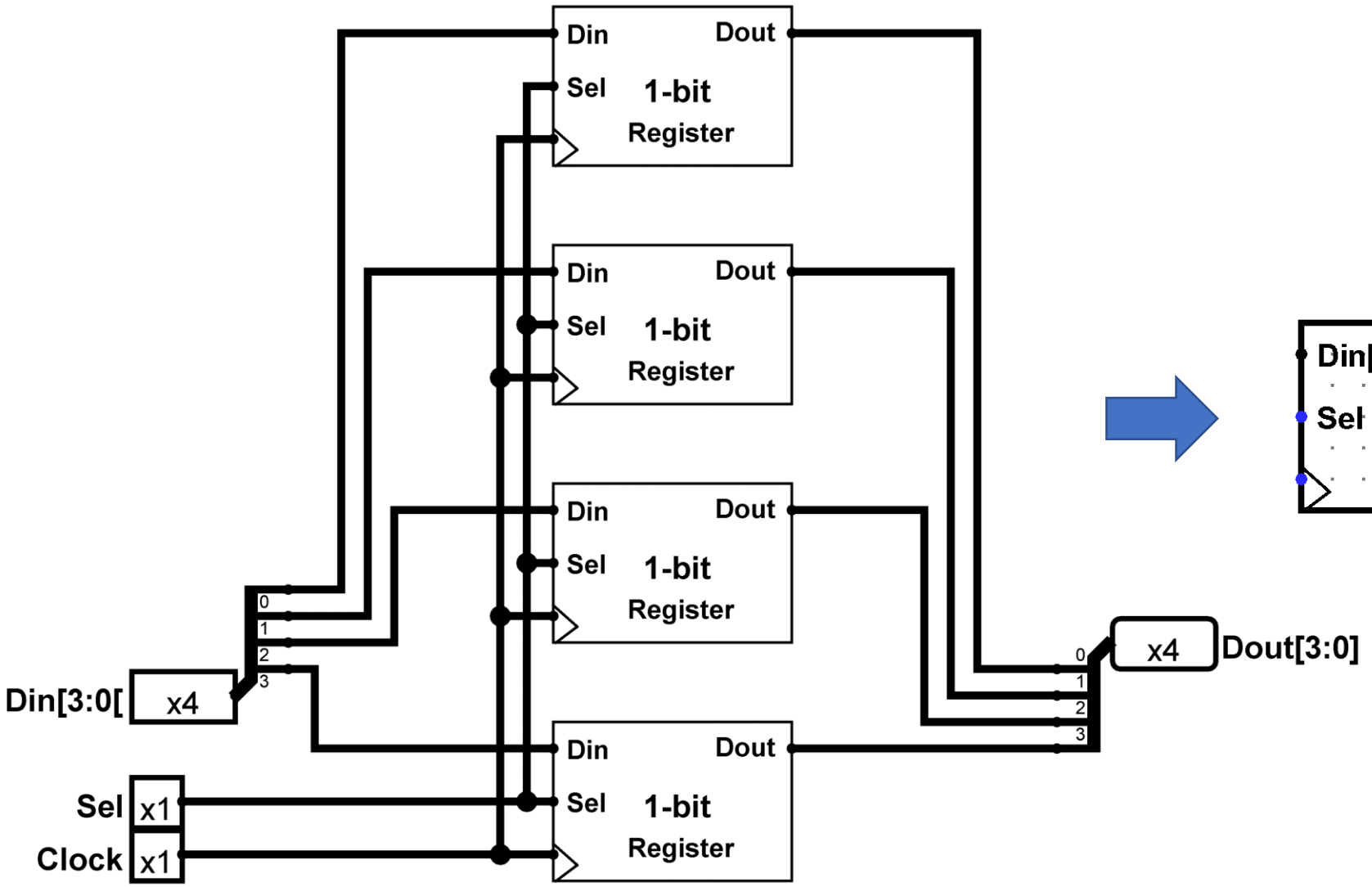
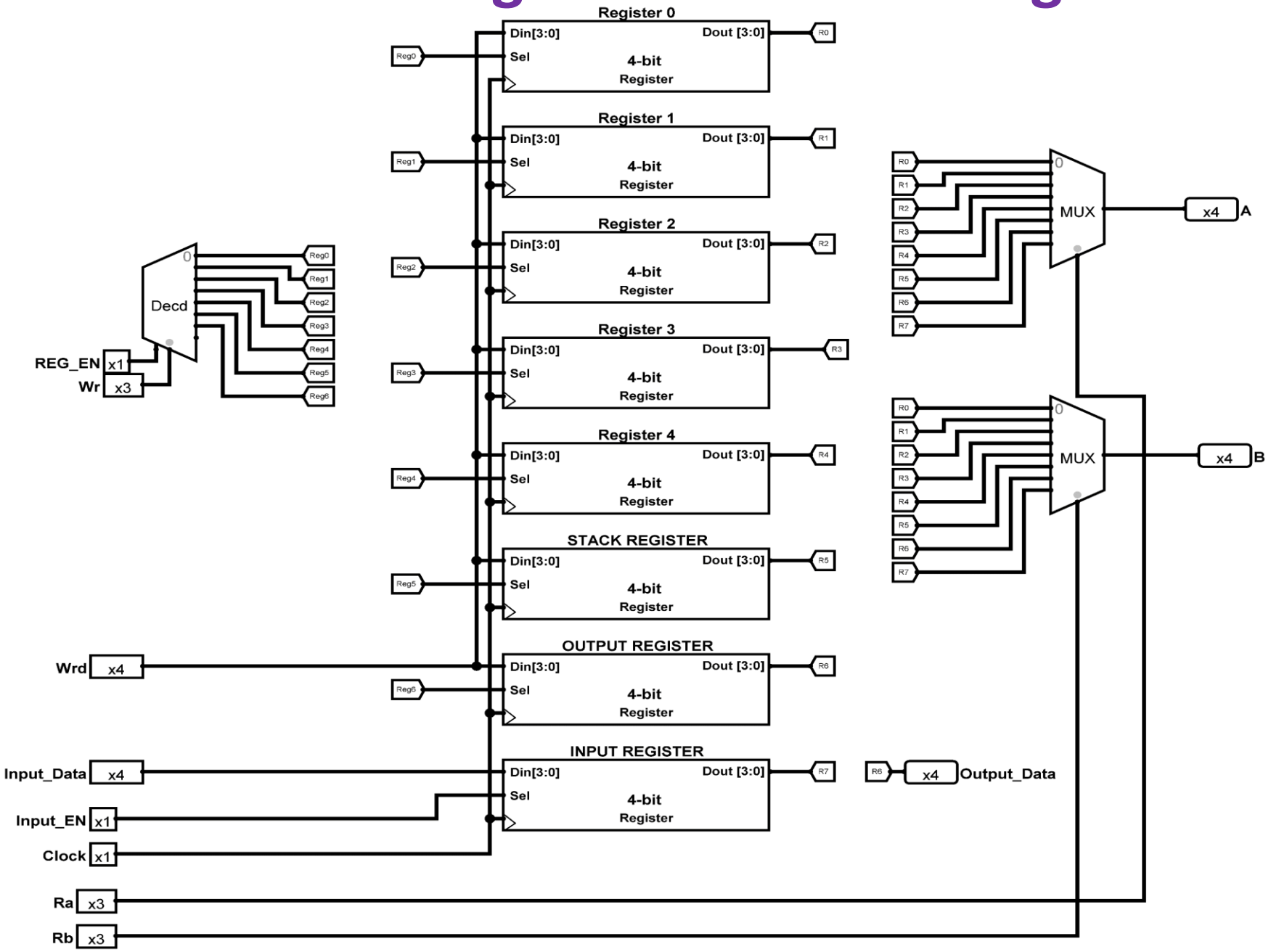


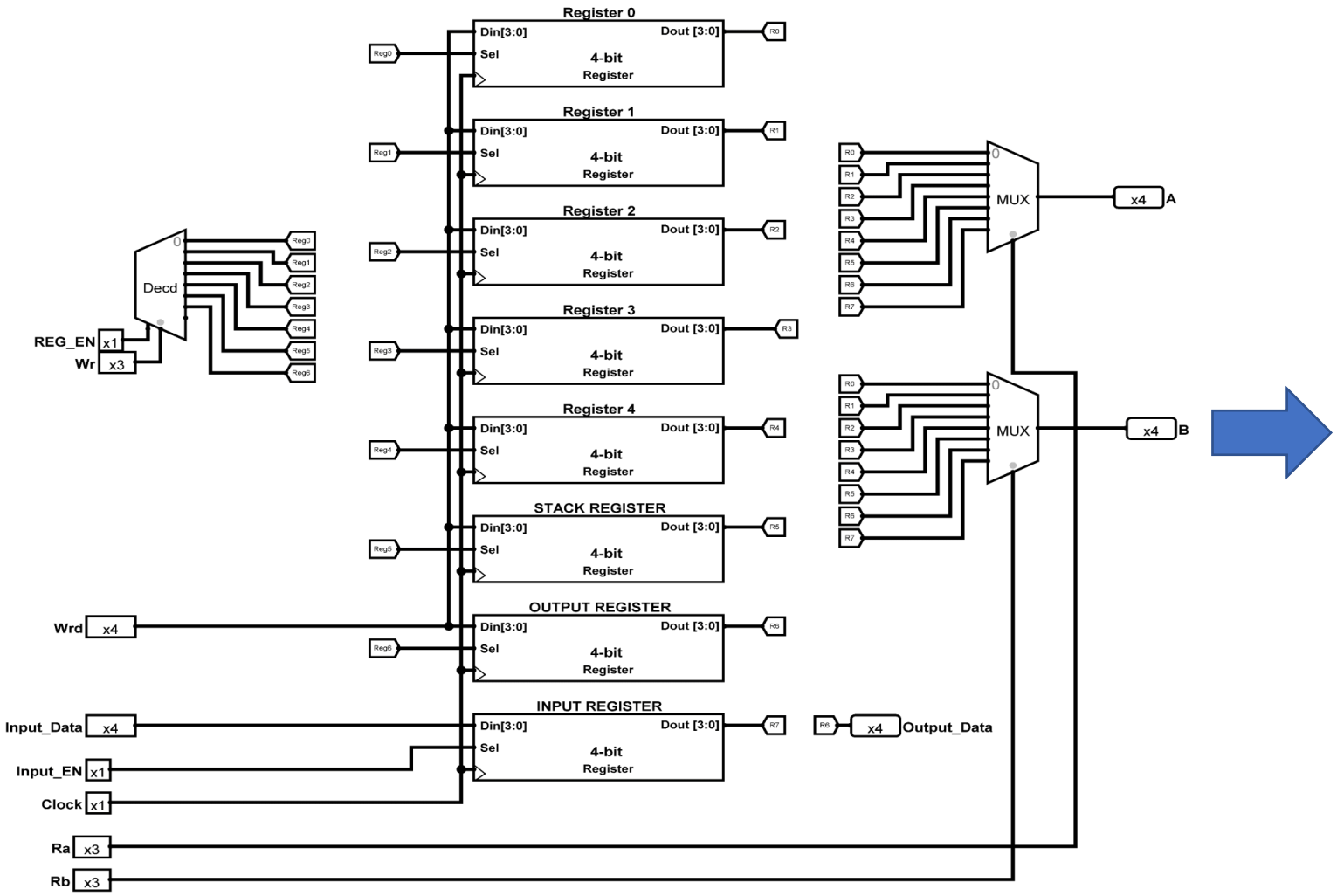
Figure: 4-bit Register Chip

4-bit Register Set with 8 Registers

4-bit Register Set with 8 Registers



4-bit Register Set with 8 Registers



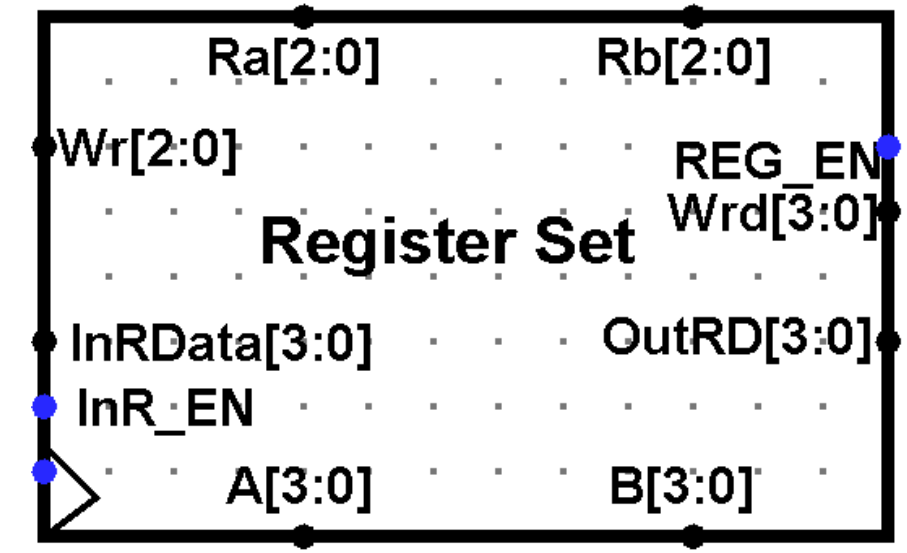
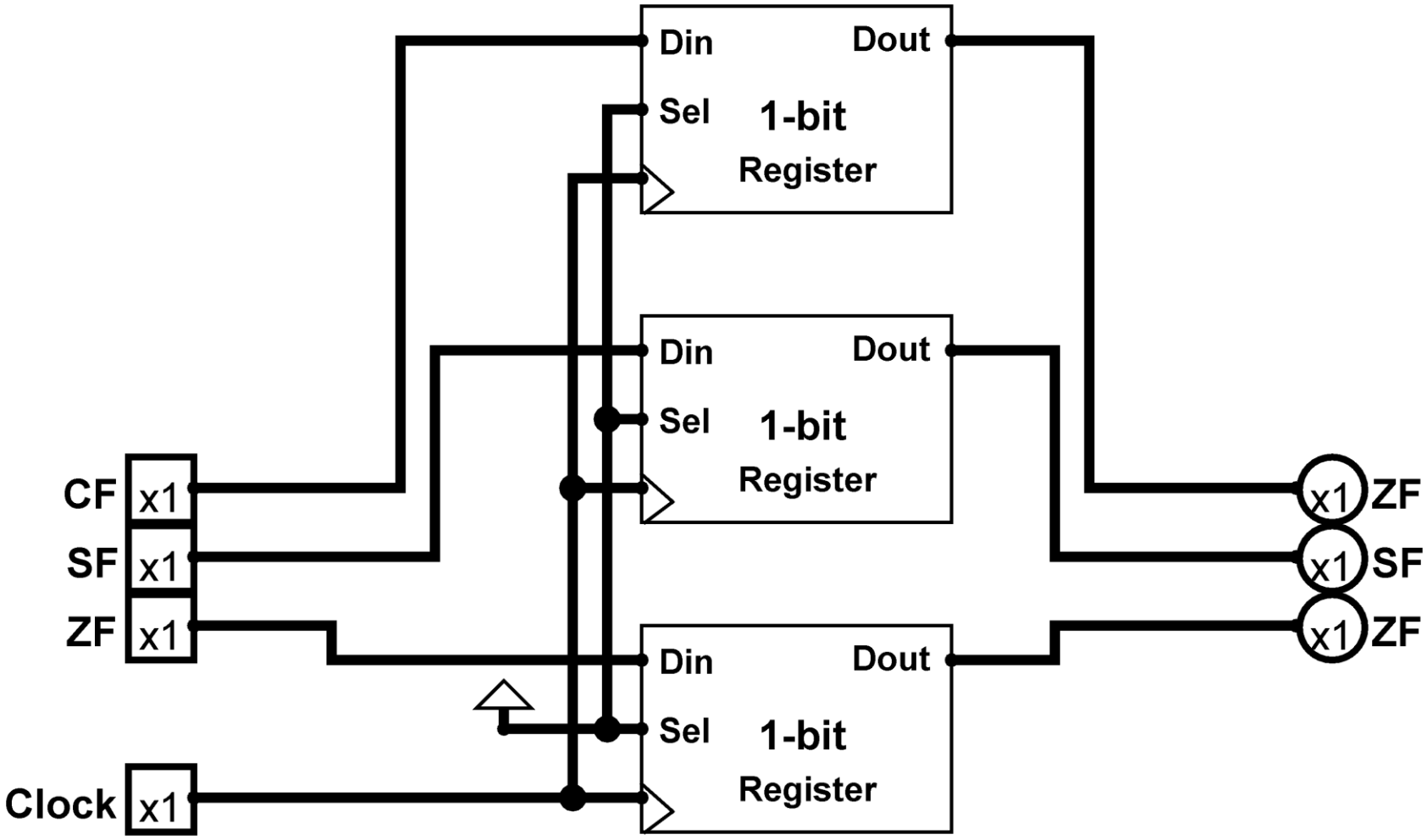


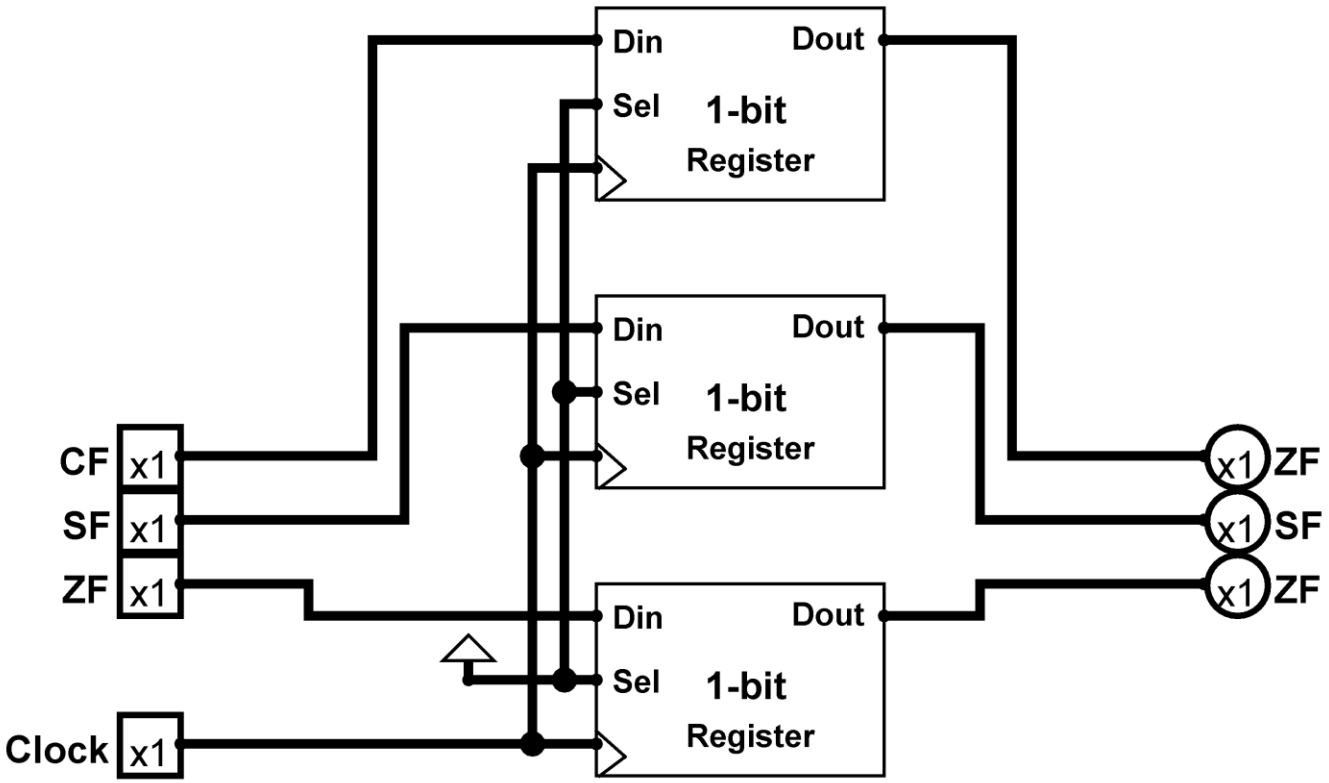
Figure: 4-bit Register Set with 8 Registers Chip

4-bit FLAG Register

4-bit FLAG Register



4-bit Register Set with 8 Registers



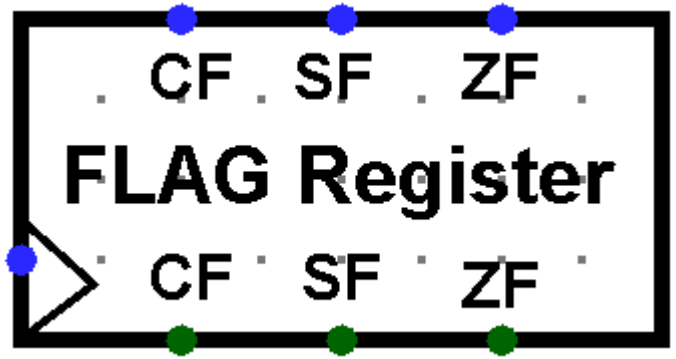


Figure: FLAG Register

Example: Register Set Design

Question: Design a 7-bit Register Set with 10 registers.

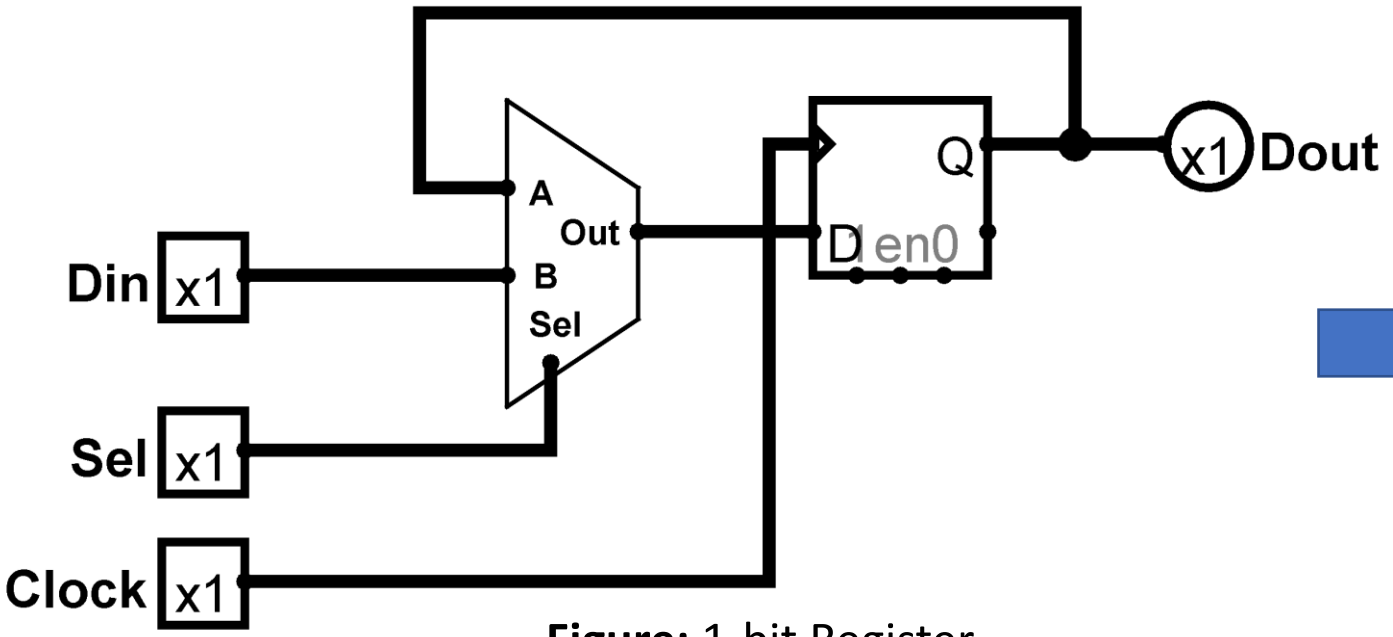


Figure: 1-bit Register

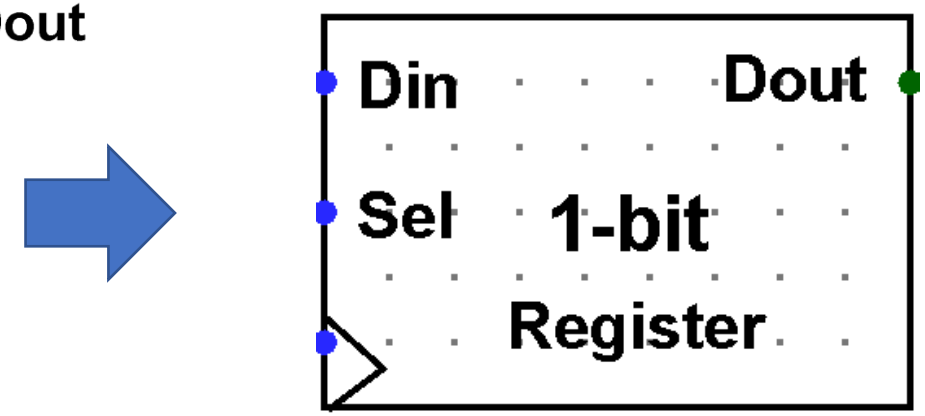


Figure: 1-bit Register Chip

Example: Register Set Design

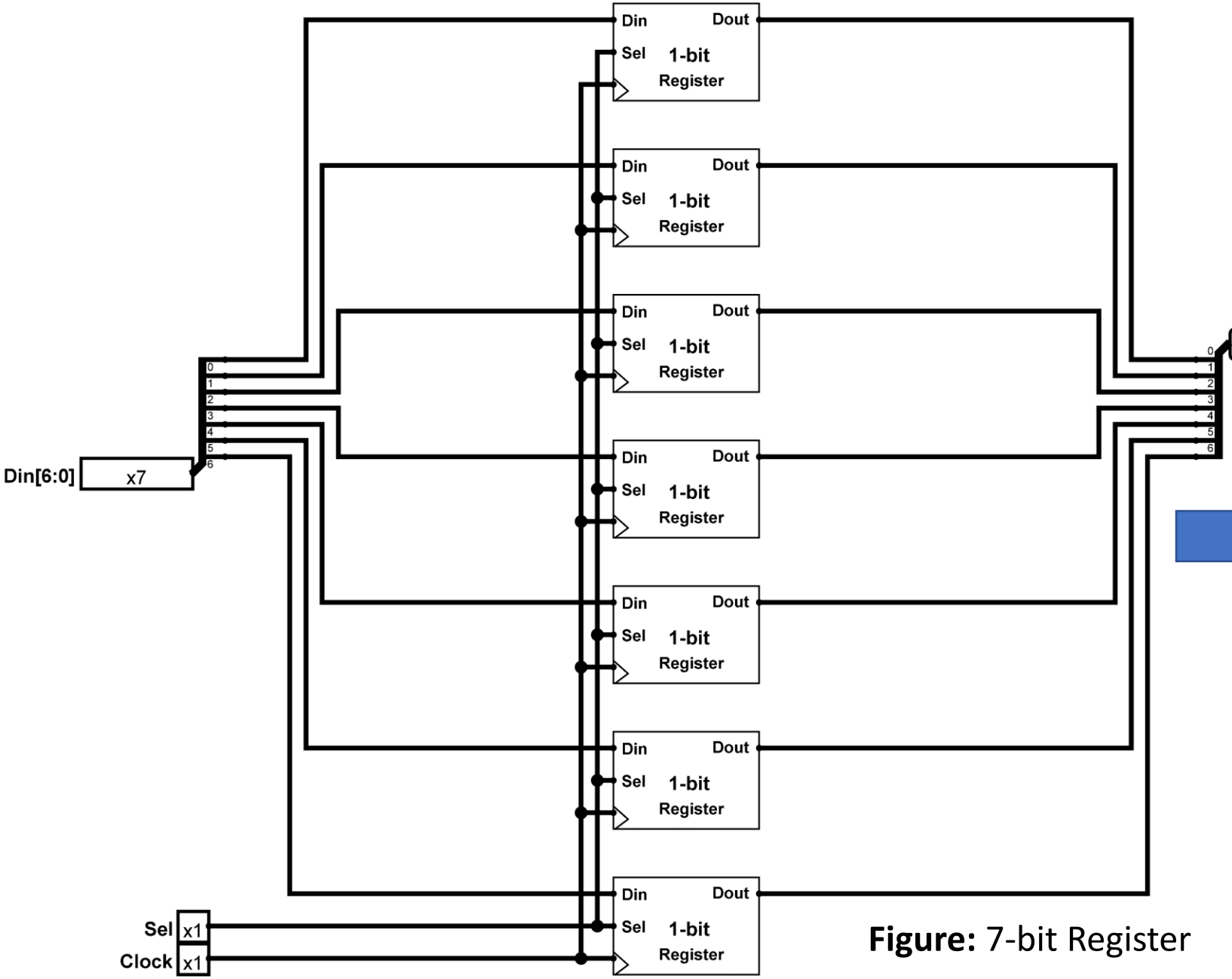


Figure: 7-bit Register

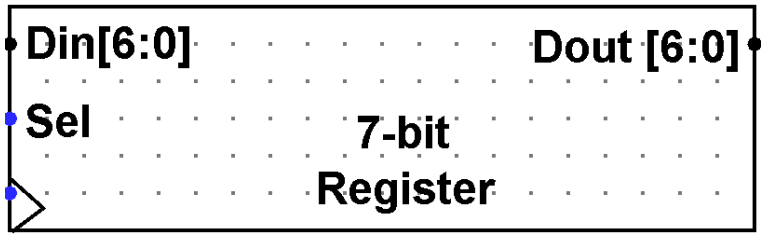


Figure: 7-bit Register Chip

Example: Register Set Design

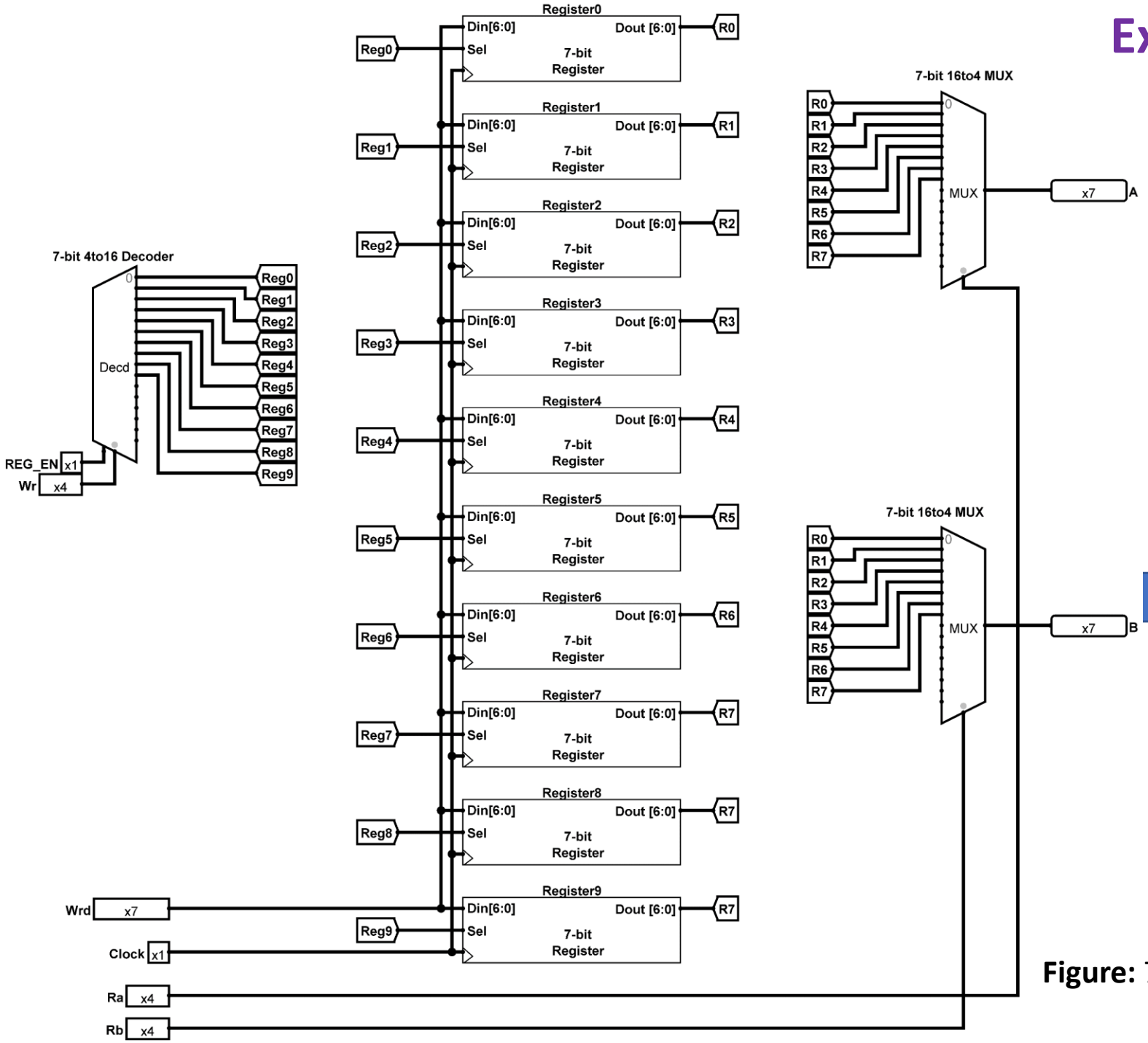


Figure: 7-bit Register Set with 10 Registers

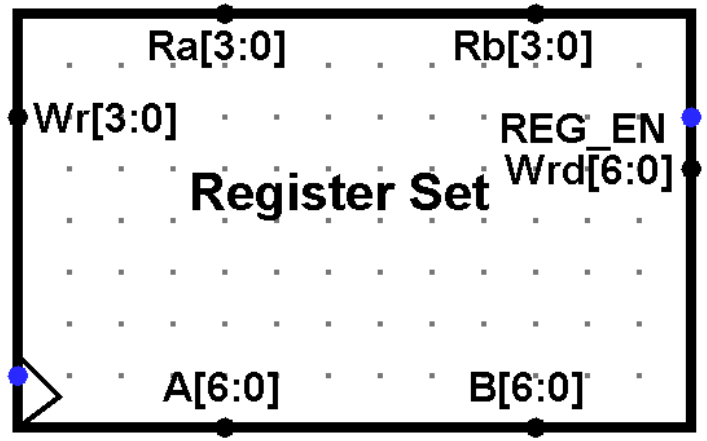


Figure: 7-bit Register Set with 10 Registers Chip

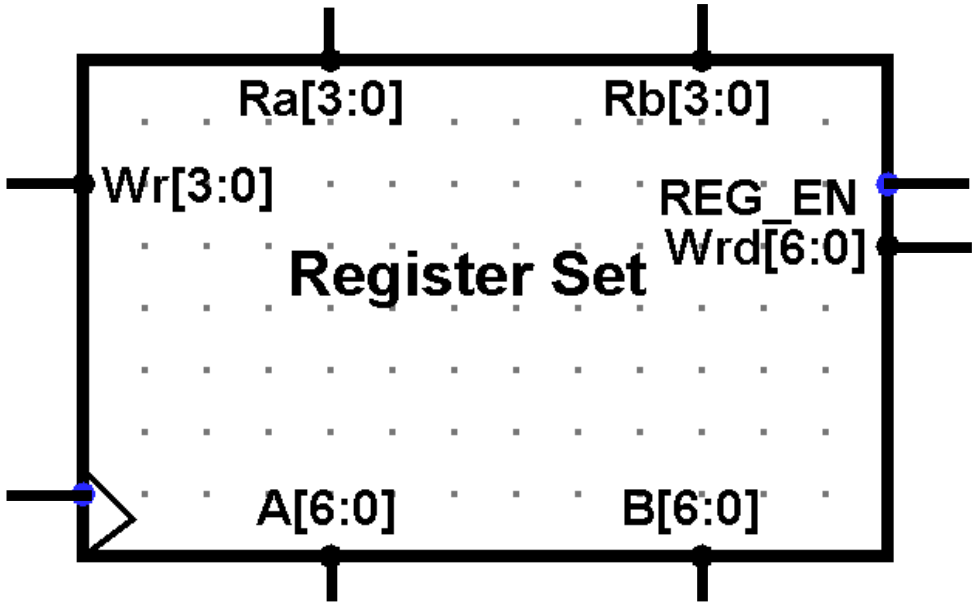
Exercises

1. Draw 2-bit/3-bit/4-bit/5-bit/6-bit/7-bit/8-bit Register.

2. Design 2-bit/3-bit/4-bit/5-bit/6-bit/7-bit/8-bit Register Set with 1/2/3/4/5/6/7/8 register/registers.

Exercises

3. Consider the following Register Set chip



	Data						
	6	5	4	3	2	1	0
R2	1	0	0	1	0	0	1
R5	1	0	0	0	1	0	1
R6	0	0	0	0	0	0	0

i. What is the size and word size of Register Set?
ii. If Ra = 0101 and Rb = 000, then what will the value of A and B?
iii. If Wr = 0010, Wrd = 1110000, REG_EN = 0, Ra = 010, Rb = 110, then what will be the value of A and B after next clock pulse?
iv. Design this Register Set chip.

Thank You ☺