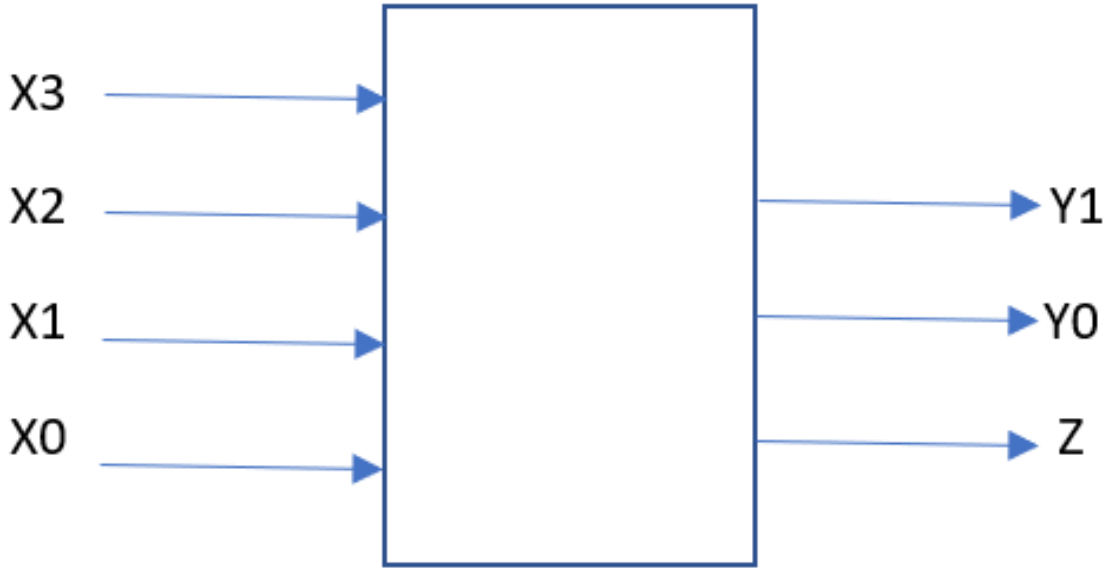
Κωδικοποιητής
Αποκωδικοποιητής
Αποπολυπλέκτης

Δυαδικός κωδικοποιητής (Encoder) (1)

- Ένας δυαδικός κωδικοποιητής κωδικοποιεί μια πληροφορία 2^n bits σε κώδικα μήκους n bits.
- Στην έξοδο του κωδικοποιητή εμφανίζεται ο αύξων δυαδικός αριθμός της εισόδου που έχει τιμή 1.
- Μόνο μια είσοδος πρέπει να έχει τιμή 1 κάθε φορά.
- Μια έξοδος z του κωδικοποιητή ελέγχει αν υπάρχει ή όχι κάποια τιμή εισόδου. Αν δεν υπάρχει τιμή σε καμία είσοδο τότε z=0, διαφορετικά z=1.

Δυαδικός κωδικοποιητής (Encoder) (2)



Κωδικοποιητής 4 σε 2

X_3	X_2	X_1	X_0	Y_1	Y_0	Z
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	1
1	0	0	0	1	1	1

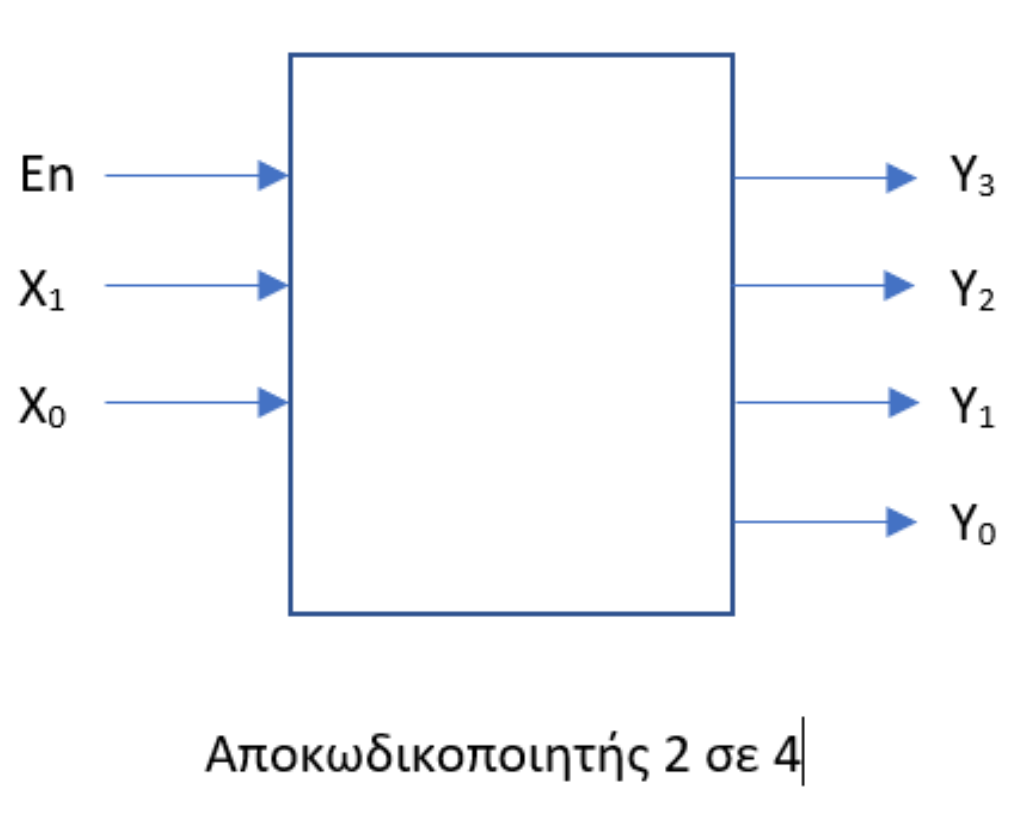
Δυαδικός κωδικοποιητής (Encoder) (3)

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY Encoder4to2 IS
    PORT (x : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
          y : OUT STD_LOGIC_VECTOR(1 DOWNTO 0);
          z : OUT STD_LOGIC);
END Encoder4to2;
ARCHITECTURE Arch OF Encoder4to2 IS
BEGIN
    PROCESS(x)
    BEGIN
        IF x(3)='1' THEN
            y <= "11";
        ELSIF x(2)='1' THEN
            y <= "10";
        ELSIF x(1)='1' THEN
            y <= "01";
        ELSE
            y <= "00";
        END IF;
    END PROCESS;
    z <= '0' WHEN x="0000" ELSE '1';
END Arch;
```

Δυαδικός αποκωδικοποιητής (Decoder) (1)

- Ένας δυαδικός αποκωδικοποιητής αποκωδικοποιεί μια πληροφορία μήκους n bits σε 2^n εξόδους.
- Μόνο μια έξοδος έχει τιμή 1 κάθε φορά.
- Ενεργοποιείται η έξοδος με αύξοντα αριθμό τον αριθμό που έχει σχηματιστεί στην είσοδο.
- Διαθέτει είσοδο ενεργοποίησης (En) των εξόδων. Αν En=0 είναι απενεργοποιημένες όλες οι έξοδοι, ενώ όταν En=1 ενεργοποιείται η έξοδος που αντιστοιχεί στον αριθμό της εισόδου.

Δυαδικός αποκωδικοποιητής (Decoder) (2)



Αποκωδικοποιητής 2 σε 4

En	X_1	X_0	Y_3	Y_2	Y_1	Y_0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

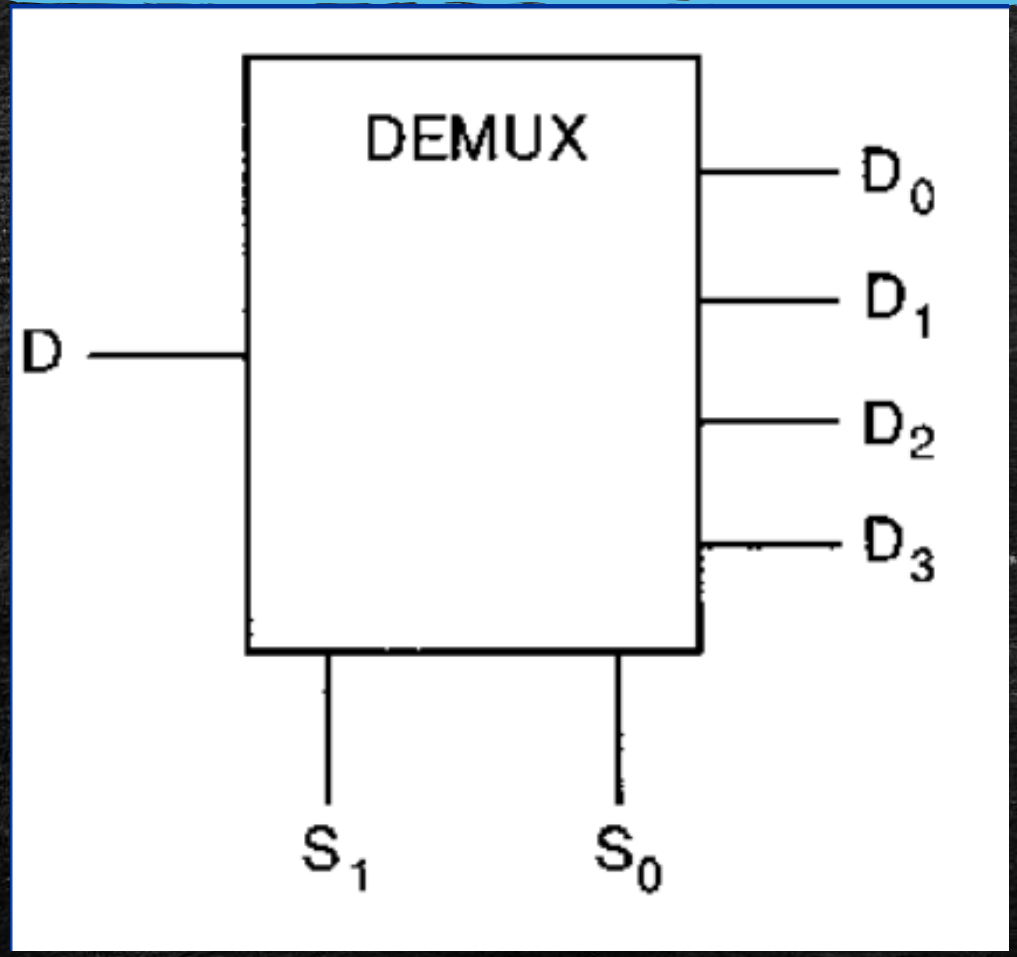
Δυαδικός αποκωδικοποιητής (Decoder) (3)

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY Decoder2to4 IS
    PORT (x : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
          en: IN STD_LOGIC;
          y : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END Decoder2to4;
ARCHITECTURE Arch OF Decoder2to4 IS
BEGIN
    PROCESS(x,en)
    BEGIN
        IF en='1' THEN
            CASE x IS
                WHEN "00" => y<="0001";
                WHEN "01" => y<="0010";
                WHEN "10" => y<="0100";
                WHEN OTHERS => y<="1000";
            END CASE;
        ELSE
            y<="0000";
        END IF;
    END PROCESS;
END Arch;
```

Αποπολυπλέκτης (Demultiplexer) (1)

- Ένας αποπολυπλέκτης μεταφέρει το σήμα από τη μια και μοναδική είσοδό του σε μια από τις εξόδους του.
- Η επιλογή της εξόδου στην οποία θα μεταφερθεί το σήμα εισόδου, γίνεται από τις εισόδους επιλογής.
- Ένας αποπολυπλέκτης με n σήματα επιλογής διαθέτει 2^n σήματα εξόδου.

Αποπολυπλέκτης (Demultiplexer) (2)



S_1	S_0	D_3	D_2	D_1	D_0
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	D	0	0
1	1	D	0	0	0

Αποπολυπλέκτης (Demultiplexer) (3)

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY Demux1to4 IS
    PORT (x: IN STD_LOGIC;
          s : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
          y : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END Demux1to4;
ARCHITECTURE Arch OF Demux1to4 IS
BEGIN
    PROCESS(x,s)
    BEGIN
        IF s="00" THEN
            y(0) <= x;
        ELSIF s="01" THEN
            y(1) <= x;
        ELSIF s="10" THEN
            y(2) <= x;
        ELSE
            y(3) <= x;
        END IF;
    END PROCESS;
END Arch;
```

Καλή συνέχεια…