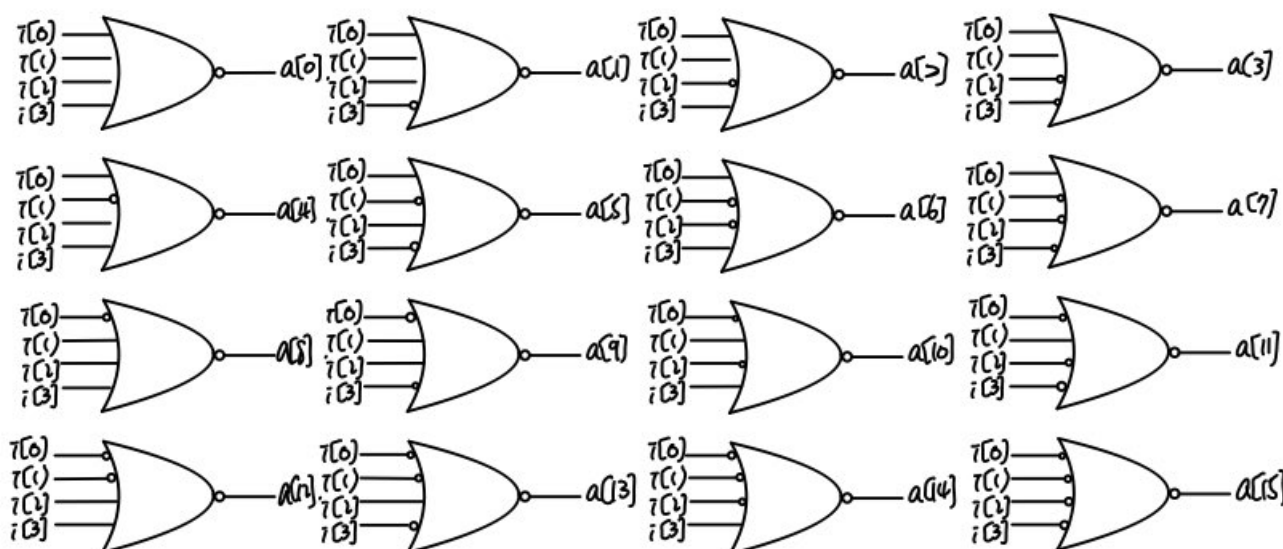


ICD 2021fall HW3 Report

EE3 b08502141 石旻翰

1 Design of Identifier

Usage: Convert input i_0, i_1, i_2, i_3, i_4 into an one-hot vector(dim = 16) which index of 1 is the decimal number of input i, respectively.

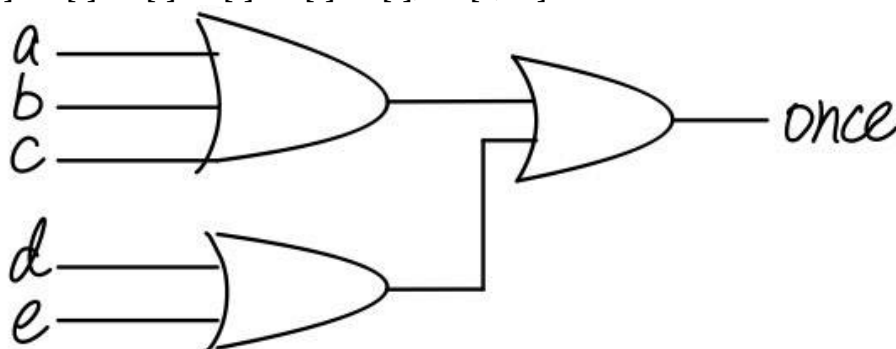


2 Design of Once, Twice, Three_times

Usage: Calculate the times of 0-15 in 5 inputs that have showed.

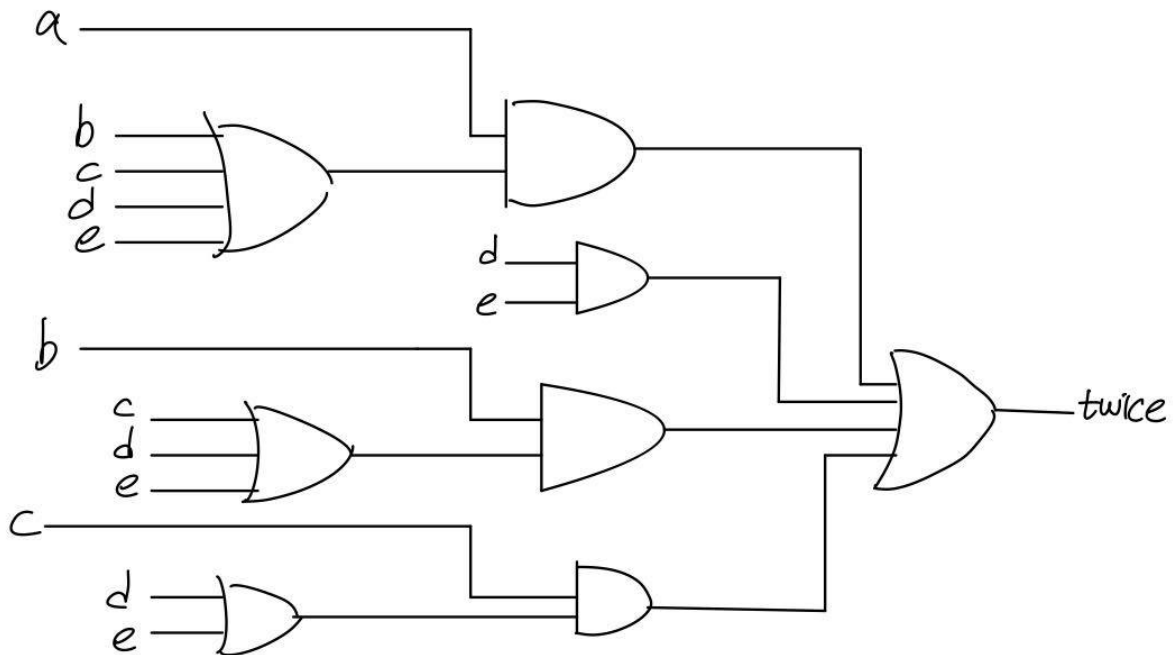
1. Once is a vector that index value means the value (0-15) showing in i_0, i_1, i_2, i_3, i_4 at least once.

$$\text{Once}[i] = a[i] + b[i] + c[i] + d[i] + e[i], i \in [0, 15]$$



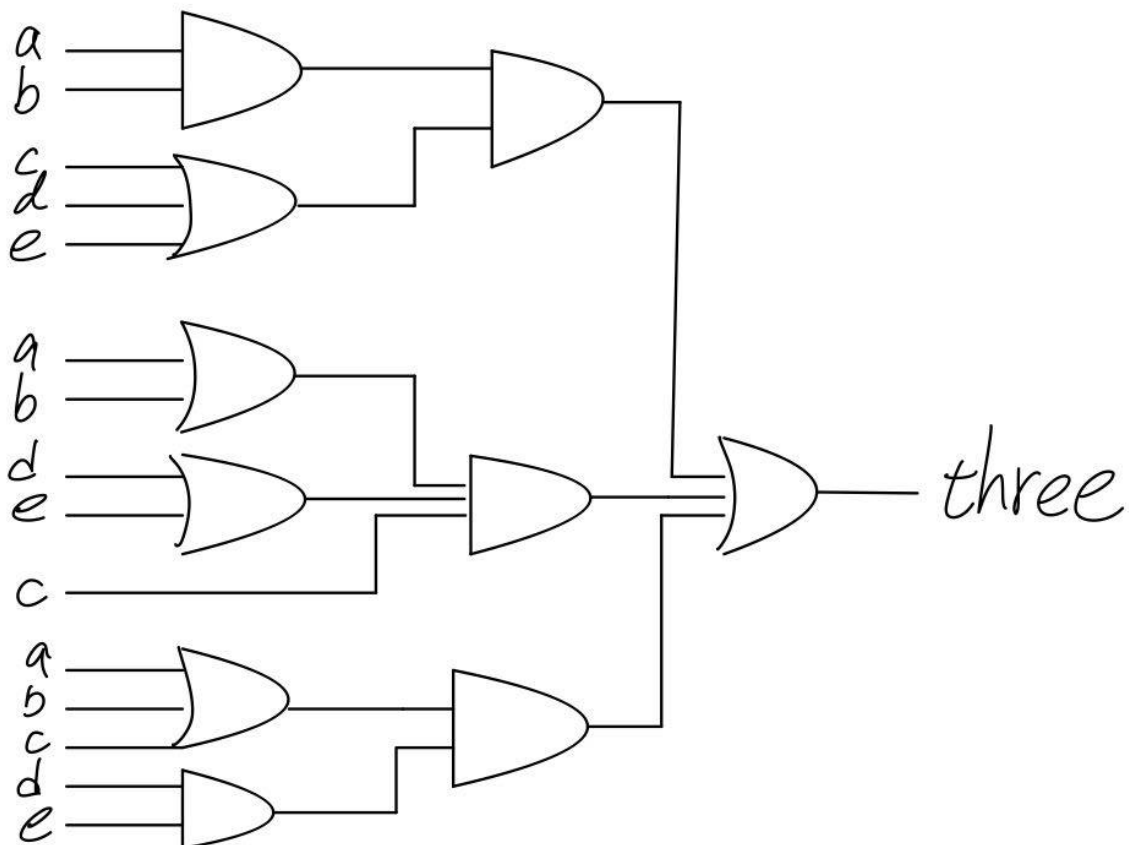
2. Twice is a vector that index value means the value (0-15) showing in i_0, i_1, i_2, i_3, i_4 at least twice.

$$Twice[i] = a[i]b[i] + a[i]c[i] + a[i]d[i] + a[i]e[i] + b[i]c[i] + b[i]d[i] + b[i]e[i] + c[i]d[i] + c[i]e[i] + d[i]e[i], i \in [0, 15]$$



3. Three is a vector that index value means the value (0-15) showing in i_0, i_1, i_2, i_3, i_4 at least three times or more.

$$Three[i] = a[i]b[i]c[i] + a[i]b[i]d[i] + a[i]b[i]e[i] + a[i]c[i]d[i] + a[i]c[i]e[i] + a[i]d[i]e[i] + b[i]c[i]d[i] + b[i]c[i]e[i] + b[i]d[i]e[i] + c[i]d[i]e[i], i \in [0, 15]$$



Note1: Once_inv, Twice_inv, Three_inv are convenient for later calculations, so they are calculated together at this part, just turn the last gates of three design into an inverted ones.(e.g.: OR->NOR, AND->NAND)

Note2: There are some differences between this gate-level diagram and my verilog program, for I have to improve the critical paths under 3ns, so I change some slow gates to some faster gates.

3 Design of Decoder

1. Special Decoder:

I found that three must be an one-hot vector, so I use the K-map and decode three[15:0](below call it as A[15:0]) into d3[3:0].

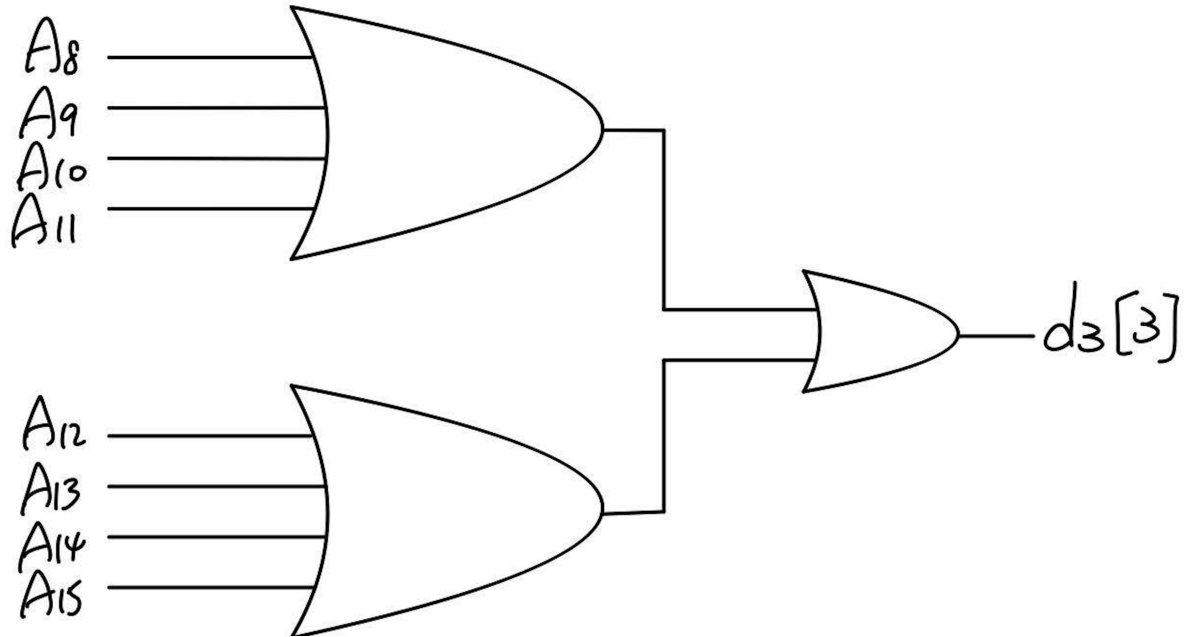
$$d3[3] = A[8] + A[9] + A[10] + A[11] + A[12] + A[13] + A[14] + A[15]$$

$$d3[2] = A[4] + A[5] + A[6] + A[7] + A[12] + A[13] + A[14] + A[15]$$

$$d3[1] = A[2] + A[3] + A[6] + A[7] + A[10] + A[11] + A[14] + A[15]$$

$$d3[0] = A[1] + A[3] + A[5] + A[7] + A[9] + A[11] + A[13] + A[15]$$

Here takes d3[3], for example, others are the same, just inputs and output need to be changed.



2. Decoder:

Convert twice[15:0] and once[15:0] into d2[3:0] and d1[3:0], respectively.

Here takes twice[15:0](below call it as B[15:0]) into d2[3:0], for example.

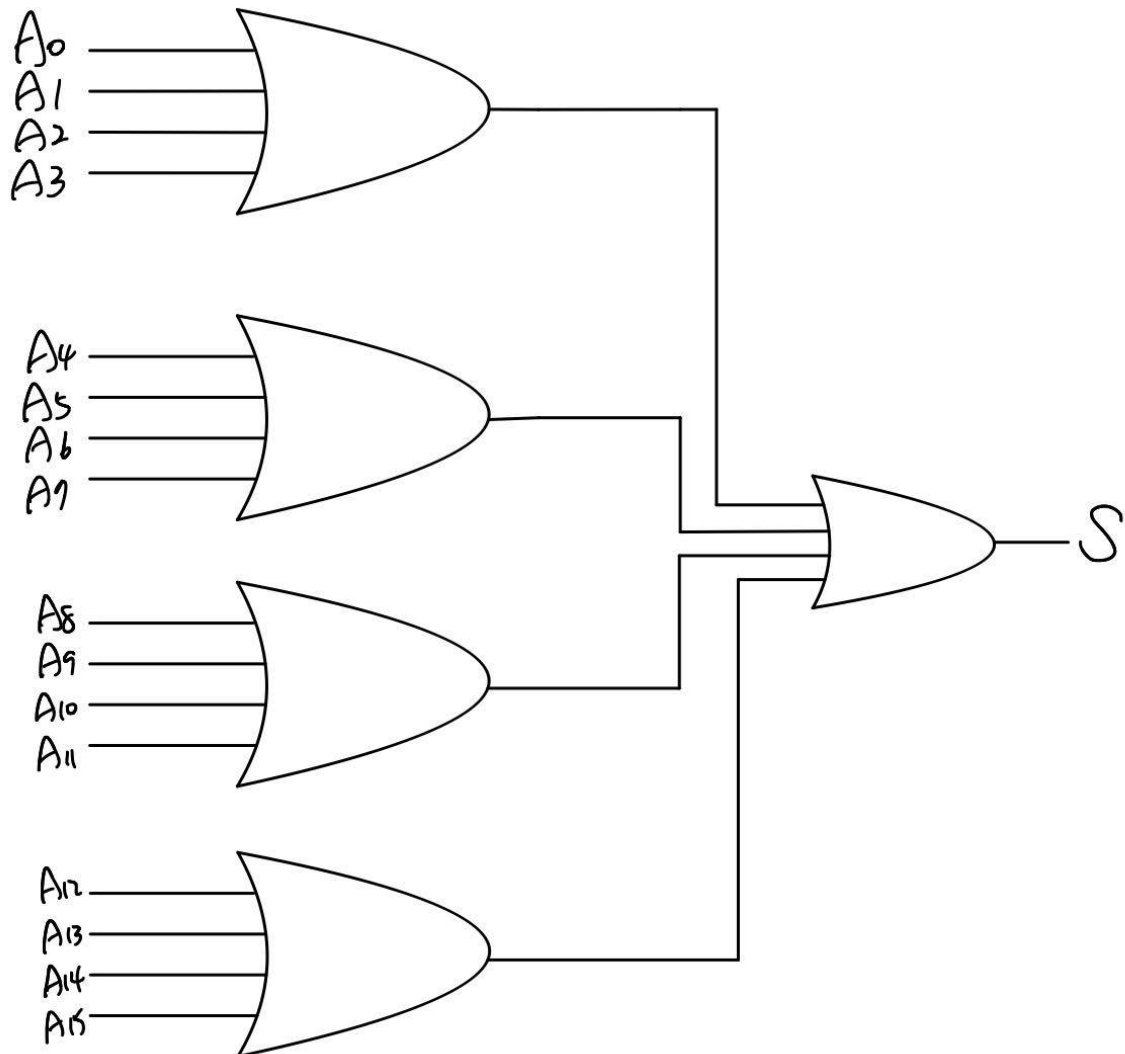
(1). Take 0-15 into 4 parts: {0-3}, {4-7}, {8-11}, {12-15}.

(2). Take {0-3} output d[3:0], for example. d[3:0] = 0011 if 3 exists (twice[3] = 1), or d[3:0] = 0010 if 2 exists and 3 DNE (twice[2] = 1 and twice[3] = 0), or d[3:0] = 0001 if 1 exists and 2 and 3 DNE, otherwise d[3:0] = 0000. And the {4-7}, {8-11}, {12-15} output c[3:0], b[3:0], a[3:0], similarly.

(3). Take a, b, c, d to generate d2[3:0]. Here I use a_e, b_e, c_e, d_e to represent that if a, b, c, d exist or not, if not exist, a_ne, b_ne, c_ne would be invert a_e, b_e, c_e, respectively. d2[3:0] = a[3:0] if a exists (a_e = 1), or d2[3:0] = b[3:0] if b exists and a DNE (b_e = 1 and a_ne = 1), or d2[3:0] = c[3:0] if c exists and a and b DNE (c_e = 1 and a_ne = 1 and b_ne = 1), otherwise d2[3:0] = d[3:0].

3. S:

Usage: Just check that whether 1 exist in twice or three or not, and it would be used in the design of Ans part.



4 Design of Ans

Similar with the part of Decoder, output $\text{mode}[3:0] = d3[3:0]$ if there is an one in three ($S3 = 1$), or $\text{mode}[3:0] = d2[3:0]$ if there is an one in twice ($S2 = 1$) and there is not an one in three ($S3 = 0$), otherwise, $\text{mode}[3:0] = d1[3:0]$ ($S2 = 0$ and $S3 = 0$).

