

מבנה מחשבים

תרגול מספר 6

DECODER(n)

- **Input:** A string $\{0,1\}^n$.
- **Output:** A string $\{0,1\}^{2^n}$.
- **Functionality:** For every string s , returns 1 in the bit which is the decimal representation of s .

Question 4.2

Question 4.2 Prove that $\text{DECODER}(n)$ is asymptotically optimal with respect to cost and delay. Namely, prove that, for every decoder G of input length n , the following hold:

$$c(G) \geq \Omega(2^n)$$
$$d(G) \geq \Omega(\log n).$$

We assume constant in-degree of every non-trivial gate with constant cost and constant delay. Trivial gates (input gates and output gates) have zero cost and zero delay.

Question 4.2 (solution)

There are $\Omega(2^n)$ outputs which are outputs of different non-trivial basic gates.

To prove this we have to show that the outputs are

- not constant
- not equal to the inputs
- not equal to another outputs

Question 4.2 (solution cont.)

- There is exactly one input string for which bit i equals 1.
→ no output is constant.
- If $y[j]=x[i]$ then the circuit is wrong, since half of the 2^n input instances $x[i]=1$ but $y[j]=1$ true only for a single input instance.
→ no output equals to any input.
- Using the same argument every output is 1 for some input and all the other outputs are 0 for that input.
→ no equal outputs.

Since there are at least 2^n different gates in every decoder its cost $\geq \Omega(2^n)$.

Question 4.2 (solution cont.)

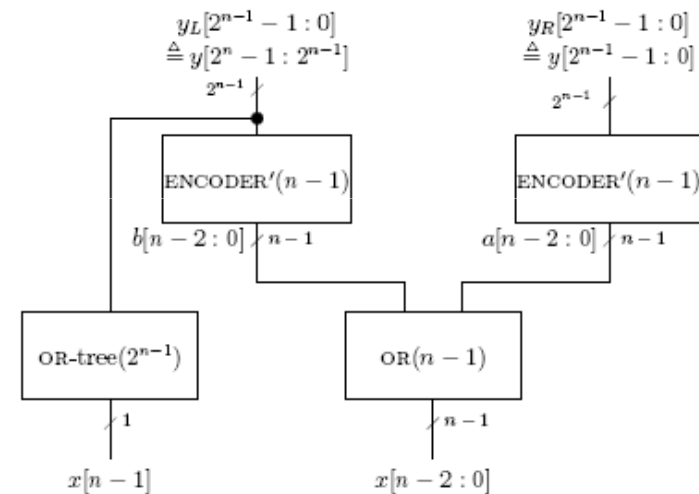
To prove the lower bound of the delay we use the cone of $y[0]$.

$y[0]$ depends on n -bits hence the delay $\geq \Omega(\log(n))$

ENCODER(n)

- **Input:** A string $\{0,1\}^{2^n}$.
- **Output:** A string $\{0,1\}^n$.
- **Functionality:** For every string s with weight 1, returns the binary representation of s .

ENCODER'(n)



Question 4.3

Question 4.3 This question deals with the cost and delay of $\text{ENCODER}'(n)$.

1. Prove that $c(\text{ENCODER}'(n)) = \Theta(n \cdot 2^n)$.
2. Prove that $d(\text{ENCODER}'(n)) = \Theta(n)$.
3. Can you suggest a separate circuit for every output bit $x[i]$ with cost $O(2^n)$ and delay $O(n)$? If so then what advantage does the $\text{ENCODER}'(n)$ design have over the trivial design in which every output bit is computed by a separate circuit?

Question 4.3(solution)

Let's n be a number of outputs and $N=2^n$ be number of inputs. We write the recurrence equations in the terms of N .

1. Let's calculate the cost $c(N)=c(\text{ENCODER}'(n=\log_2 N))$
 $c(N=2)=0$.
 $c(N>2)=2c(\text{ENCODER}'(n-1))+c(\text{OR-TREE}(2^{n-1}))+c(\text{OR}(n-1))$
 $c(N)=2c(N/2)+(N/2-1)+(\log_2 N)=2c(N/2)+\Theta(N)$.
 Applying Master theorem we have $c(N)=\Theta(N \log N)=\Theta(n2^n)$

2. Let's calculate the delay $d(N)=d(\text{ENCODER}'(n=\log_2 N))$
 $d(N=2)=0$.
 $d(N>2)=\max\{d(\text{ENCODER}'(n-1)) + d(\text{OR}), d(\text{OR-TREE}(N/2))\}$
 $d(N)=\max\{d(N/2) + \text{tpd}(\text{OR}), \Theta(\log(N))\}$
 The solution for this recurrence is $d(N)=\Theta(\log N)=\Theta(n)$.

Definition:

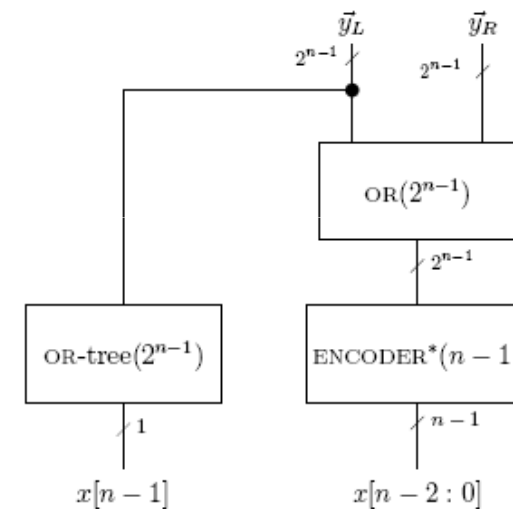
A Boolean function $f:\{0,1\}^n \Rightarrow \{0,1\}$ is *monotone* if $x \geq y \Rightarrow f(x) \geq f(y)$

(where $x \geq y$ means : for every i $x_i \geq y_i$).

Claim:

$f:\{0,1\}^n \Rightarrow \{0,1\}$ is *monotone iff* f can be implemented by a combinational circuit that contains only **AND-gates** and **OR-gates**.

ENCODER*(n)



Question 4.7

Question 4.7 The designs $\text{ENCODER}'(n)$ and $\text{ENCODER}^*(n)$ lack inverters, and hence are monotone circuits. Is the Boolean function ENCODER_n a monotone Boolean function? Suppose that G is an encoder and is a monotone combinational circuit. Suppose that the input y of G has two ones (namely, $\text{wt}(y) = 2$). Can you immediately deduce which outputs of G must equal one?

It is a trivially monotone function since for every two valid inputs x and y the following is false: $x \geq y$ and $y \geq x$.

Let y be Bitwise-OR of two valid inputs y_1 and y_2 . Since we are talking about a monotone function for input y we will get an output $z \geq w$ where w is a Bitwise-OR of $w_1 = f(y_1)$ and $w_2 = f(y_2)$.

MULTIPLEXER

Input: $D[n - 1 : 0]$ and $S[k - 1 : 0]$ where $k = \lceil \log_2 n \rceil$.

Output: $Y \in \{0, 1\}$.

Functionality:

$$Y = D[\langle \bar{S} \rangle].$$

MULTIPLEXER decoder based implementation

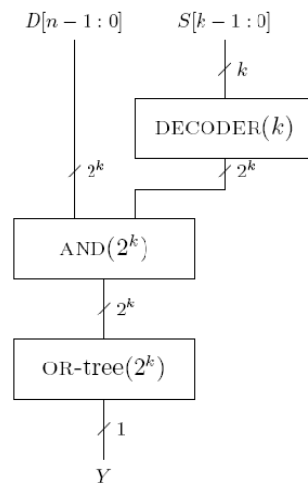


Figure 5.1: An $(n:1)$ -MUX based on a decoder ($n = 2^k$).

MULTIPLEXER decoder based implementation

1. Prove the correctness of the design.

We should prove that $Y = D[\langle \bar{S} \rangle]$

$$\begin{aligned}
 Y &= \text{OR}(\text{AND}(D[0], \text{DECODER}[0]), \dots, \text{AND}(D[n-1], \text{DECODER}[n-1])) \stackrel{\text{DECODER}[i]=1 \iff \langle S \rangle = i}{=} \\
 &= \text{OR}(\text{AND}(D[0], 0), \dots, \text{AND}(D[\langle \bar{S} \rangle], 1), \dots, \text{AND}(D[n-1], 0)) \stackrel{\text{AND}(x,0)=0, \text{AND}(x,1)=x}{=} \\
 &= \text{OR}(0, 0, \dots, 0, D[\langle \bar{S} \rangle], 0, \dots, 0) \stackrel{\text{OR}(0,x)=x}{=} D[\langle \bar{S} \rangle]
 \end{aligned}$$

MULTIPLEXER decoder based implementation

2. Analyze the cost and delay of the design.

$$c(MUX) = c(DECODER(\log n)) + n \cdot c(AND) + c(OR-TREE) = \Theta(n) + n \cdot \Theta(1) + \Theta(n) = \Theta(n)$$

$$d(MUX) = d(DECODER(\log n)) + d(AND) + d(OR-TREE) = \Theta(\log \log n) + \Theta(1) + \Theta(\log n) = \Theta(\log n)$$

MULTIPLEXER decoder based implementation

3. Prove that the cost and delay of the design are asymptotically optimal.

Cost: the circuit has $n + \log(n)$ inputs which all have an influence on the output. Therefore the $|Cone(Y)| = n + \log(n)$.

From every input there is a path to the output, hence every input connected to a gate.

Since a gate's in-degree is constant we get a lower bound the number of gates: $\Omega(n + \log n) = \Omega(n)$

Delay: Again, due to *Cone* considerations

$$Delay = \Omega(\log |Cone|) = \Omega(\log(n + \log n)) = \Omega(\log n)$$

Our design meets the following lower bounds and therefore it is asymptotically optimal.