

# ECE1388 Project 2

## Circuit and Layout Design for Optimum Performance: 32-Word Register Decoder

Consider a row decoder used to select one word within a 32-word register file, where each word is 64 bits wide. An example of such a decoder, for the case when it uses 4 stages of static CMOS logic to drive each word line, is shown in Fig. 1(a) (IP stands for ‘Input’).

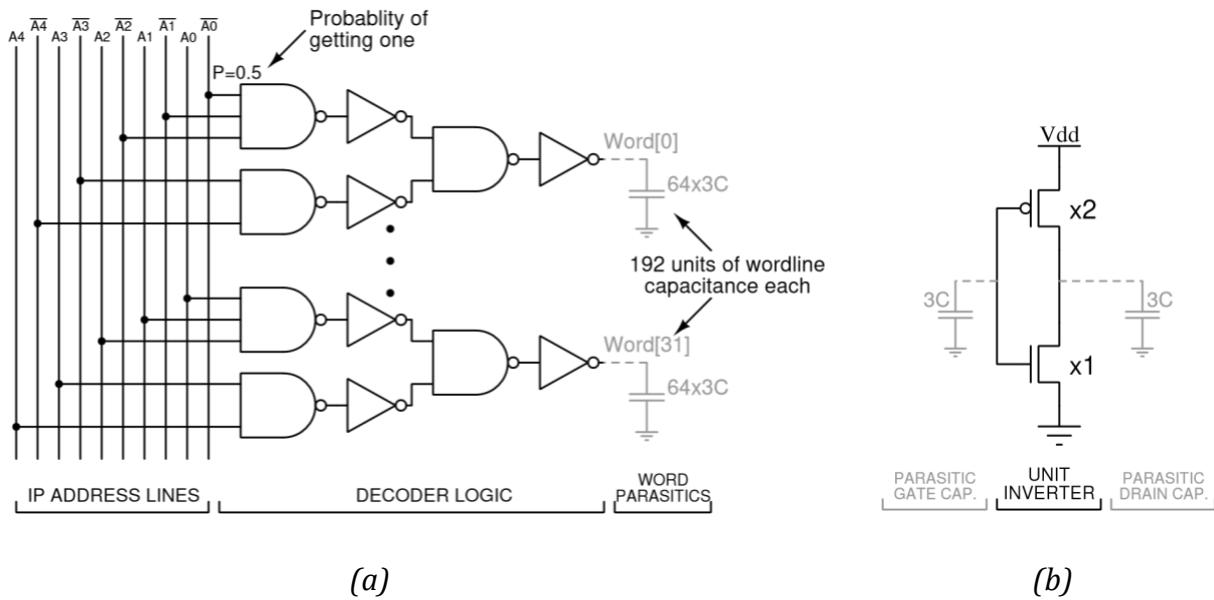


Figure 1. (a) An example of a 4-stage decoder architecture. (b) A unit-size inverter definition.

- Assume a unit-size inverter is defined as shown in Fig. 1 (b).
- Assume:  $V_{dd} = 1V$ ,  $\lambda = 30nm$ , temperature is  $25^\circ C$ .
- Use shaping inverters for input stimuli in all simulations.
- Consult Chapters 4, 5, 6, and 7 of the textbook and the lecture notes.

## 1. Circuit Characterization and Performance Estimation: Logical Effort and Gate Sizing

- a) Estimate the propagation delay of 6 different decoder slice designs, with the number of stages ranging between 2 to 6 stages. Use only inverters and NAND gates with no more than 5 inputs.

The decoder logic for each of the six designs is shown in Fig. 2 (OP stands for ‘Output’).

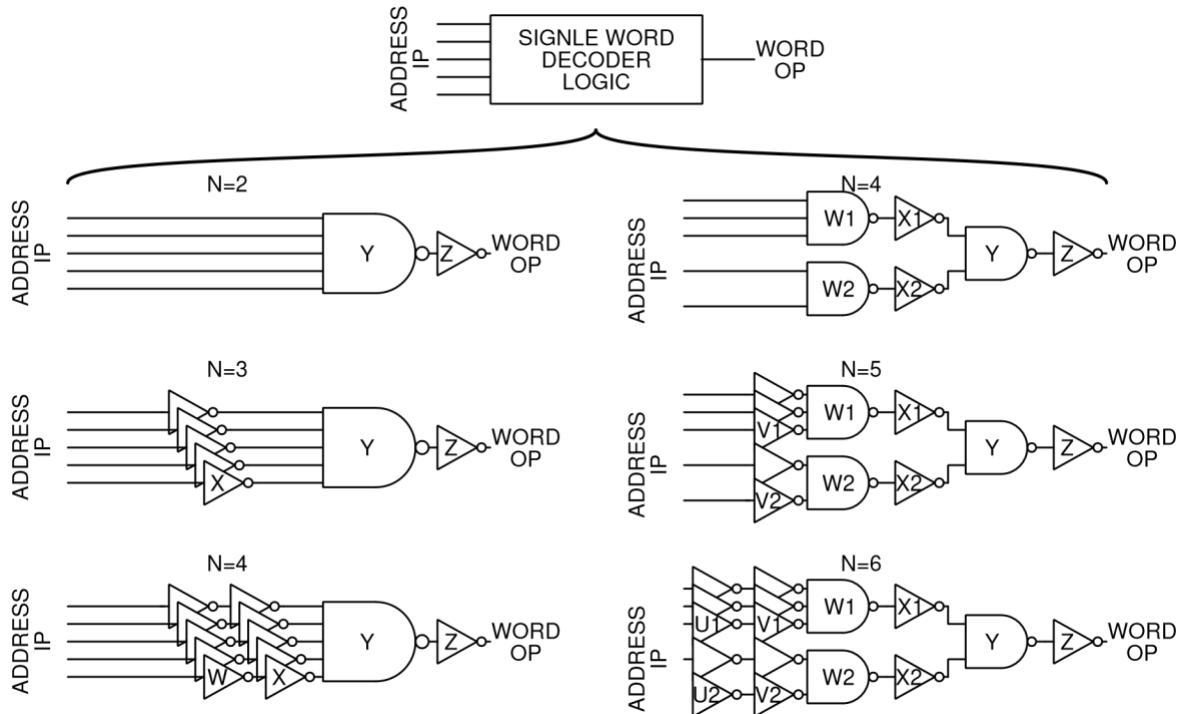


Figure 2: Six decoder architectures investigated in this project.

Base the design on the example in Section 4.5.3 of the textbook. Make the same design assumptions as in the textbook. Your design should not include the 64-bit registers, but you will need to consider their presence for the capacitive loading calculations. Include your calculations in the report.

The specifications for the design can be summarized as follows:

- A 32-word register file, where each word is 64 bits wide
- Each input address line can drive up to 20 unit-size inverters<sup>1</sup> (as shown in Fig. 1 (b))
- Each register bit adds a load of three unit-size transistors to the word line, as shown in Fig. 1(a) (accounting for two unit-size access transistors plus some wire capacitance)
- Both true and complementary versions of the address bits A[4:0] are available

---

<sup>1</sup> The textbook uses a unit-size transistor, but for the ease of calculation, use an inverter instead

- b) Populate all empty cells in Table 1 below with the performance characteristics for each decoder architecture:

Table 1: Comparison of decoder designs

Design Architecture	N	G	P	D	Relative size of different stages					
					Z	Y	X	W	V	U
NAND5-INV							--			
INV-NAND5-INV							--			
INV-INV-NAND5-INV								--		
NAND2/3-INV-NAND2-INV								--		
INV-NAND2/3-INV-NAND2-INV									--	
INV-INV-NAND2/3-INV-NAND2-INV										

where

- $N$  is the number of stages,
- $G$  is the path logical effort,
- $P$  is the path parasitic delay, and
- $D$  is the path delay

Denote the smallest delay in **bold** font.

- c) Calculate the (average) dynamic power dissipation for the fastest design assuming the input address is changing in a *rolling*<sup>2</sup> fashion, incrementing monotonically from all zeros to all ones in  $32\mu\text{sec}$ . Assume all 32 input addresses are equiprobable. You will need to consider the gate and diffusion capacitance for the power calculation.
- d) Use a stick diagram to estimate the area in lambda and microns of the fastest decoder. Use the logical effort method to size all logic gates (using relative sizes for each stage).

---

<sup>2</sup> Input address changes in the format 0, 1, 2 ... 31. A new input address is applied at every  $1\mu\text{s}$ .

## 2. Schematic Entry and Transistor-Level Simulation

- a) Design Cadence Composer schematic and (self-describing) symbol views for inverters and NAND gates based on the transistor sizing obtained in item 1 above for the fastest decoder. Also, design the top-level schematic and symbol views for this decoder. Keep the design hierarchical and modular. You may use the standard cells from *tcbn65gplus* library.

Figure 3 shows an example of a *partially complete* top-level decoder schematic that uses arrayed instantiation of the single-word decoder logic symbol and bus naming convention.

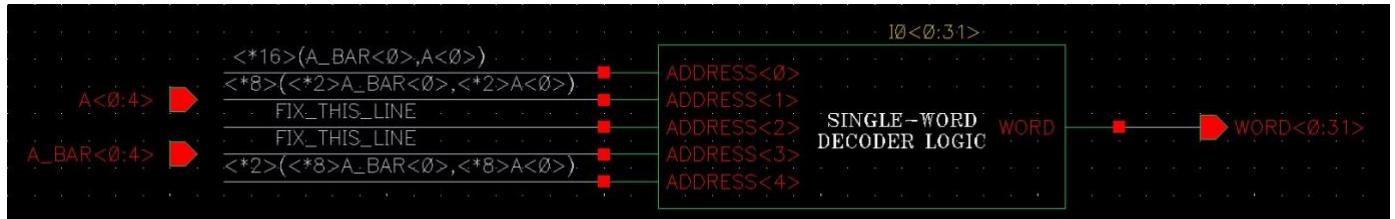


Figure 3: Top-level decoder schematic.

- b) Validate the circuit delay value obtained in question 1 through simulation. Explain any observed differences.
- c) Obtain the simulated value for the total power dissipation of the circuit in 2 (a) averaged across all combinations of the address. (hint: see equation 5.3 in textbook).
- d) Obtain the simulated value for the static power dissipation of the circuit in 2 (a) averaged across all combinations of the address. (hint: consider what happens when one address is held constant).
- e) Validate the circuit dynamic power dissipation value obtained in question 1, by subtracting the result of item 2 (d) from the result of item 2 (c). Explain any differences.

## 3. Layout Design and Post-Layout Verification.

- a) Layout the fastest decoder and perform DRC and LVS verification. Minimize the layout area while following standard cell layout style. Label all I/O pins for each cell and all global nodes. Include the single decoder level and top-level layout printouts and top level LVS report file in your report.
- b) Simulate the extracted view to obtain the values for the delay as in question 2 (b) and for power dissipation as in questions 2 (c-e).
- c) Compare the results for the delay and the average dynamic power dissipation for your calculations (where applicable), schematic simulation, and extracted view simulation. Show your comparison results in the form of a table.