Below is a concise walkthrough of how that slide (and its scribbled notes) derive the circuit for a 3-bit input ($X_2X_1X_0$) whose 3-bit 2's-complement output ($Y_2Y_1Y_0$) equals "input 2."

---

1. Truth Table Construction

Because the input is only 3 bits (0 through 7 in decimal), you list each input and compute (input − 2) in 3-bit 2's-complement:

Decimal Input	$X_2X_1X_0$	(input − 2) in decimal	(input − 2) in 3-bit 2's-comp	$Y_2Y_1Y_0$
0	000	-2	110	110
1	001	-1	111	111
2	010	0	000	000
3	011	1	001	001
4	100	2	010	010
5	101	3	011	011
6	110	4	100	100
7	111	5	101	101

From this table:

- $Y_2 = 1$ for input decimal $\{0, 1, 6, 7\}$
- $Y_1 = 1$ for input decimal $\{0, 1, 4, 5\}$
- $Y_0 = 1$ for input decimal $\{1, 3, 5, 7\}$

---

2. Setting Up the K-Maps

A Karnaugh map for 3 inputs can be arranged in a 2×4 or 4×2 grid. A common layout is:

- Columns labeled by (X_2, X_1) in Gray code: 00, 01, 11, 10
- Rows labeled by $X_0 = 0$ and $X_0 = 1$

You create three separate K-maps, one for each output bit Y_2, Y_1, Y_0. Below is a conceptual sketch for Y_2. The same grid pattern is repeated for Y_1 and Y_0.

```
        X2 X1
        00   01   11   10
X0=0     ?    ?    ?    ?
X0=1     ?    ?    ?    ?
```

(a) K-Map for Y_2

- Mark a 1 in the cells corresponding to $(X_2X_1X_0)$ in $\{000, 001, 110, 111\}$.
- Notice that these cells form two 2×1 vertical "blocks" that combine to a larger grouping if your K-map arrangement is standard.
- The minimal Sum of Products (SOP) from that grouping is:

  $$Y_2 = (X_2'X_1') + (X_2X_1).$$

  Equivalently, $Y_2 = 1$ when X_2 and X_1 are equal—often called the "equivalence" function $X_2 \leftrightarrow X_1$.

(b) K-Map for Y_1

- From the truth table, Y_1 is 1 for $\{000, 001, 100, 101\}$.
- All of those have $X_1 = 0$, so the simplified form is:

  $$Y_1 = X_1' \; = \; \neg X_1.$$

(c) K-Map for Y_0

- From the truth table, Y_0 is 1 for $\{1, 3, 5, 7\}$ in decimal, i.e., whenever $X_0 = 1$.
- So, immediately:

  $$Y_0 = X_0.$$

---

3. Final Expressions

Putting it all together, the final minimal SOP expressions for "input − 2" in 3-bit 2's-complement are:

$$
\begin{aligned}
Y_2 &= X_2' X_1' \;+\; X_2 X_1, \\
Y_1 &= X_1', \\
Y_0 &= X_0.
\end{aligned}
$$

These three outputs correctly yield (input 2) for any 3-bit input $X_2X_1X_0$.

Key Takeaways

1. **Truth Table**: Enumerate all inputs (0–7), then subtract 2 in 3-bit 2's-complement.
2. **K-Maps**:
   - Each output bit gets its own 3-input K-map.
   - Label in Gray code so adjacent cells differ by only one bit.
3. **Simplification**:
   - Identify the largest adjacent groupings of 1-cells in each K-map.
   - The resulting minimal SOP reveals the neat patterns:
     - Y_2 is an equivalence function of (X_2, X_1).
     - Y_1 is simply $\neg X_1$.
     - Y_0 is simply X_0.

Hence, the scribbles you see on the slide show how to fill the K-maps (grouping 1-cells) and arrive at those nicely reduced Boolean expressions.