Below is a concise walkthrough of how that slide (and its scribbled notes) derive the circuit for a 3‐bit input whose 3‐bit 2’s‐complement output 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 in 3‐bit 2’s‐complement:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decimal Input |  | in decimal | in 3‐bit 2’s‐comp |  |
| 0 | 000 |  | 110 | 110 |
| 1 | 001 |  | 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:

* for input decimal
* for input decimal
* for input decimal

**2. Setting Up the K‐Maps**

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

* Columns labeled by in Gray code: 00, 01, 11, 10
* Rows labeled by and

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

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

**(a) K‐Map for**

* Mark a 1 in the cells corresponding to in .
* 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:Equivalently, when and are equal—often called the “equivalence” function .

**(b) K‐Map for**

* From the truth table, is 1 for .
* All of those have , so the simplified form is:

**(c) K‐Map for**

* From the truth table, is 1 for in decimal, i.e., whenever .
* So, immediately:

**3. Final Expressions**

Putting it all together, the final minimal SOP expressions for “” in 3‐bit 2’s‐complement are:

These three outputs correctly yield for any 3‐bit input .

**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:
     + is an equivalence function of .
     + is simply .
     + is simply .

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.