



INITIALIZE

Strictly speaking, the Digi-Comp II does not have an "INITIALIZE" function- it is a task that the human operator must perform, before most computations, to prepare the wooden "circuitry" for use.

The INITIALIZE procedure consists of three steps:

1. Be sure that T1, T2, T3, T4, T5, and T6 are to the left
2. Be sure that CF1 is to the left and that CF2 is to the right
3. Set Flip-Flops D1, D2, and D3 in the Distributor to the right

The default, "initialized" position of each of these flip-flops and switches is marked on the deck of the DCIIFE with a heavy dark line. So in practice, initializing just consists of making sure that each of these components is pointing in the direction indicated by the line. You will find that the distributor will need to be reset after most operations, but that the other parts (CF1, CF2, T1-T6) tend to need setup only at the beginning of a computation session.

Again, the INITIALIZE procedure only prepares the wooden circuitry for use. It does not clear the registers nor select the next operation that will be performed.

CLEAR

The CLEAR operation clears the Accumulator Register; i.e., it sets each bit in the register to zero. Procedure:

- Make sure that there is at least one ball in the input guide.
- Set the CLEAR switch to ON
- Set the other function switches* (MULTIPLY, COUNT, and COMPLEMENT) to OFF
- Set the Automatic-Manual ("AM") switch to MAN (manual)
- START

The CLEAR switch directs the ball into a special channel on the lower deck, where it rolls *beneath* the Accumulator Register. There is a "half flip-flop" below each flip-flop in the Accumulator that switches each 1 to a 0, and leaves each 0 alone. Since the entire operation is completed with a single ball, the "manual" mode is used, so that it does not trigger any additional balls when the operation completes.

COUNT

At the heart of the Digi-Comp II is a binary counter. You can use this fact, with the COUNT operation, to count the number of balls in the input guide.

- INITIALIZE
- Set the Accumulator Register to 0000000 (e.g., with a CLEAR operation)
- Put the desired number of balls into the input guide
- Set the COUNT switch to ON
- Set the other function switches (MULTIPLY, CLEAR, and COMPLEMENT) to OFF
- Set the AM switch to AUTO
- START

Each ball enters the Accumulator at the top, effectively adding one to the 1's place of the register. If the flip-flop contains a 0, it flips to 1. If it contains a 1, it flips to 0 and carries: the ball rolls down to the next flip-flop and adds a 1 at the 2's place (i.e., binary 10's), and the process continues—each ball triggering the next after it finishes—until the supply of balls is exhausted. The final count is displayed as the number in the Accumulator. Binary numbers in the registers are always read bottom to top, so a count of "1" would read 0000001, bottom to top.

Note. The Accumulator can only hold numbers as large as 127 (binary 1111111). If your count should exceed this—i.e., overflow to have a 1 in the 128's place (binary 1xxxxxx)—the OVERFLOW switch setting will determine the behavior. If it is set to "HALT," the operation will end upon overflow. If it is set to "RUN," the operation will continue after overflow.

Addition and/or Multiplication

In modern terms, the Digi-Comp II has a single-process "multiply-accumulate" operation, that can add, multiply, or both. The Multiply and Addition operations are special cases of this operation. The operation could be described as $C' = (A \times B) + C$, **where** A is the initial value of the MQ Register, B is the value of the Memory Register, C is the initial value of the Accumulator Register, and C' is the final value of the Accumulator Register. If the initial value of C is zero, then we just have $C' = (A \times B)$ a pure multiplication. If instead the initial value of A is zero, then we just have $C' = (A \times 1) + C$, or simply $C' = A + C$; a pure addition.

Procedure:

- INITIALIZE, and load the input guide with balls
- Set the MQ Register to the value A
- Set the Memory Register to the value B
- Set the Accumulator Register to the value C
- Set the MULTIPLY switch to ON
- Set the other function switches (CLEAR, COUNT and COMPLEMENT) to OFF
- Set the AM switch to AUTO
- START

The result will appear in the Accumulator Register when the Digi-Comp II halts.

Example: Multiply 3 x 5

For multiplication, we begin with a value of zero in the Accumulator. The two operands (3 and 5) go in the MQ Register and the Memory Register. (Either operand can go in either register)

Procedure:

- INITIALIZE, and load the input guide with balls
- Set the Accumulator Register to 0000000 (e.g., with a CLEAR operation)

- Set the MQ Register to the value 3 (binary 011)
- Set the Memory Register to the value 5 (binary 0101)
- Set the MULTIPLY switch to ON
- Set the other function switches (CLEAR, COUNT and COMPLEMENT) to OFF
- Set the AM switch to AUTO
- START

When the Digi-Comp II halts (after 13 balls), the result will appear in the Accumulator: 15 (binary 0001111).

Example: Add 10 + 23

For addition, we begin with a value of 1 in the MQ register, and the other two operands go into the Memory Register and Accumulator Register. Since the Memory Register only stores values up to 15 (binary 1111), 10 goes in the Memory and 23 goes in the Accumulator.

Procedure:

- INITIALIZE, and load the input guide with balls
- Set the MQ Register to the value 1 (binary 001)
- Set the Accumulator Register to 23 (binary 0010111)
- Set the Memory Register to the value 10 (binary 1010)
- Set the MULTIPLY switch to ON
- Set the other function switches (CLEAR, COUNT and COMPLEMENT) to OFF
- Set the AM switch to AUTO
- START

When the Digi-Comp II halts (after 5 balls), the result will appear in the Accumulator Register: 33 (binary 0100001)

Aside: How many balls do you need in the input guide for the full computation? It turns out that you need $4A + 1$ balls, where **A** is the initial value in the MQ Register.

Multiplication is performed by looped addition. That is to say, the value in the Memory Register is added to the value in the Accumulator **A** times, where **A** is the initial value of the MQ register. Each addition cycle takes 4 balls to complete. On the first ball, and every four balls after that, the Digi-Comp II checks the value in the MQ Register. If the value is nonzero, the value in the MQ register is decremented and a four-ball addition cycle begins, where that first ball (that performed the checked the MQ value and decremented its value) is the first of the four balls used in that addition cycle. On the other hand, if the value of the MQ register is zero, the computation halts: The ball goes down the left-hand side of the Digi-Comp II and does not trigger the start lever.

So, for a simple addition (where the initial MQ value- "**A**" in our terminology - is 1), only one addition cycle (using four balls) executes. The next ball checks and finds

the value of the MQ register to be zero, and halts the computation, so exactly five balls are needed. As the MQ register can hold a value as large as 7 (binary 111), the largest number of balls that will ever be required for a Multiply-Accumulate operation is $4 \times 7 + 1 = 29$.

Complements, Negative Numbers, and Subtraction

The other major operation of the Digi-Comp II is the COMPLEMENT function, which calculates the **two's complement** of a number in the Accumulator. The two's complement is a binary-number representation of a negative number, for working with "signed" integers in binary. For example, the two's complement of the number 3 represents "-3" in binary. This binary "-3" is the correct number such that if you should add 3 to it, you will get zero (with an overflow). Together with the addition function, this ability to calculate the two's complement allows us to perform subtraction.

The two's complement of a binary number is calculated by first inverting every bit in the number (swapping each 0 for a 1 and each 1 for a 0) and then adding 1 to the result. (The intermediate result with each bit inverted is called the **one's complement**.) The Digi-Comp II automates this process:

Procedure:

Calculate the two's complement of a number **C**

- INITIALIZE, and load the input guide with at least four balls
- Set the Accumulator Register to the value **C**
- Set the COMPLEMENT switch to ON
- Set the other function switches (CLEAR, COUNT and MULTIPLY) to OFF
- Set the AM switch to AUTO
- START

The result will appear in the Accumulator Register when the Digi-Comp II halts, which will be after four balls.

Aside: How does the COMPLEMENT operation work? The COMPLEMENT operation is managed by a simple four-stage state machine comprised of flip-flops CF1 and CF2, which takes four balls to complete.

The first ball is directed by CF1 into a special channel on the lower deck of the Digi-Comp II, where it travels beneath T1 - T6, the **accumulator mode switches**. On the lower deck, there are six flip-flops located beneath and connected to T1 - T6. As the ball rolls from one flip-flop to the next, it switches each to the opposite side, which also switches T1 - T6 to the opposite side. When it finishes, the accumulator mode switches T1 - T6 will point to the right.

When T1 - T6 are pointing to the left (their default, initialized state), the Accumulator Register is a **counter**. A single ball added to the top will increase the value of the register by one. But when they point to the right, the Accumulator Register is an **inverter**. A single ball added to the top will flip every bit from 1 to 0

or vice versa. The second ball does just this: It is directed by flip-flop CF1 into the top of the Accumulator, and flips every bit, thereby calculating the one's complement. As this ball exits, it is directed by the COMPLEMENT switch to Flip-Flop CF2 before pulling the start switch and triggering the next ball. We have already flipped every bit in the Accumulator. To complete the COMPLEMENT operation (which calculates the two's complement), we still need to add one to the value in the Accumulator. To do that, we first need to change it back from being an inverter to being a counter. The third ball is directed by CF1 into the channel on the lower deck, and changes each of the accumulator mode switches (T1 - T6) back from right to left, thereby changing the Accumulator back from being an inverter to being a counter. Finally, the fourth ball is directed by CF1 into the top of the accumulator, and adds one to the value shown there. As this ball exits, it is directed by the COMPLEMENT switch to Flip-Flop CF2. The last time that a ball went through CF2 (the second ball), it was directed to the left. This time, it is directed to the right, where it does not trigger the start lever, and the operation halts.

Example: Find the Two's Complement of 3

Procedure:

- INITIALIZE, and load the input guide at least four balls
- Set the Accumulator Register to the value 3 (binary 0000011)
- Set the COMPLEMENT switch to ON
- Set the other function switches (CLEAR, COUNT and MULTIPLY) to OFF
- Set the AM switch to AUTO
- START

When the Digi-Comp II halts (after 4 balls), the result will appear in the Accumulator Register: -3 (binary 1111101).

Example: Subtraction: calculate 15 - 10

Subtraction is a two-step process, which we formulate as adding a negative number to a positive one. In other words, we calculate $15 + (-10)$. First, calculate the two's complement of 10, and then add 15 to it.

Procedure:

- INITIALIZE, and load the input guide at least nine balls
- Set the Accumulator Register to the value 10 (binary 0001010)
- Set the COMPLEMENT switch to ON
- Set the other function switches (CLEAR, COUNT and MULTIPLY) to OFF
- Set the AM switch to AUTO
- START

After 4 balls the intermediate result will appear in the Accumulator: -10 (binary 1110110)

Next, we need to add 15 to that result. For addition, we use the multiply/accumulate function:

- Set the MQ Register to the value 1 (binary 001)
- Set the Memory Register to the value 15 (binary 1111)
- Set the MULTIPLY switch to ON
- Set the COMPLEMENT to OFF
- Set the OVERFLOW switch to RUN
- START

When the Digi-Comp N halts (after 5 balls), the result will appear in the Accumulator Register: 5 (binary 0000101).

Division

Place the dividend in the Accumulator Register, and the divisor in the Memory Register. Run the Complement operation to find the two's complement (negative) of the dividend. Then, repeatedly add the divisor to the Accumulator, until it overflows (i.e., reaches zero). The number of addition operations required is the quotient. The repeated additions are performed like a multiplication operation; the number of addition cycles is counted on the MQ register. It is necessary to invert each digit as we read it from the MQ register when doing division. This is because we are using the MQ register as an up-counter, rather than its usual role as a down-counter.

Example: Calculate $30/5$

Procedure:

- INITIALIZE, and load the input guide
- Set the Accumulator Register to the value 30 (binary 0011110)
- Set the COMPLEMENT switch to ON
- Set the other function switches (CLEAR, COUNT and MULTIPLY) to OFF.
- Set the AM switch to AUTO
- START
- Wait for the COMPLEMENT operation to halt

After 4 balls the intermediate result appears in the Accumulator Register: -30 (binary 1100010).

- Set the COMPLEMENT switch to OFF
- Set the MULTIPLY switch to ON
- Set the Memory Register to the value 5 (binary 0101)
- Set the MQ Register to the value 0 (binary 000)*
- Set the OVERFLOW switch to HALT
- START

When the Digi-Comp II halts, the result will appear in the MQ Register: 6 (binary 110). Remember that MQ register values are inverted when doing division.

NEW OPERATIONS CHECK LIST

INITIALIZE

1. Be sure that T1, T2, T3, T4, T5, and T6 are to the left.
2. Be sure that CF1 is to the left and that CF2 is to the right.
3. Set Flip-Flops D1, D2, and D3 in the Distributor to the right.

OPERATION	MQ REGISTER SETTING	MEMORY REGISTER SETTING	ACCUMULATOR REGISTER SETTING	SWITCH SETTINGS					
				MULTIPLY	CLEAR	COUNT	COMPLE- MENT	OVER- FLOW	AM
COUNT			0000000	OFF	OFF	ON	OFF	RUN	AUTO
CLEAR				OFF	ON	OFF	OFF	RUN	MAN.
ADD (A + B)	001	A	B	ON	OFF	OFF	OFF	RUN	AUTO
MULTIPLY (A × B)	A	B	0000000	ON	OFF	OFF	OFF	RUN	AUTO
SUBTRACT (A - B)	001	A		OFF	OFF	OFF	ON	RUN	AUTO
				ON	OFF	OFF	OFF	RUN	AUTO
DIVIDE (A / B)	000* (111, before inverting)	B	A	OFF	OFF	OFF	ON	RUN	AUTO
				ON	OFF	OFF	OFF	RUN	AUTO