

Multiplikation:

16-Bit Variante

```
ram(0) <= "0110" & "000000000000";      -- IN A X
ram(1) <= "0010" & "000000010100";      -- Store in M(20) X
ram(2) <= "0110" & "000000000000";      -- In A Y
ram(3) <= "0010" & "000000010101";      -- Store in M(21) Y
ram(4) <= "0111" & "000000010001";      -- Springe zur Ergebnis ausgabe wenn 0
ram(5) <= "0001" & "000000010110";      -- Load M(22)
ram(6) <= "0011" & "000000010100";      -- Addiere Acc + M(20)
ram(7) <= "0010" & "000000010110";      -- Store M(22)
ram(8) <= "0001" & "000000010101";      -- Load M(21)
ram(9) <= "0100" & "000000010111";      -- Sub A -M(23)
ram(10) <= "1001" & "000000000011";     -- Jump ram(3)
```

```
ram(17) <= "0001" & "000000010110";     --Load M(22)
ram(18) <= "0110" & "000000000001";     -- Out
ram(20) <= "0000000000000000";          -- M(20) <= 0  X
ram(21) <= "0000000000000000";          -- M(21) <= 0  Y
ram(22) <= "0000000000000000";          -- M(22) <= 0  Ergebnis
ram(23) <= "0000000000000001";          -- M(23) <= 1  1
```

--8-Bit Variante --

```
ram(0) <= "00000000" & "100" & "00000"; -- IN A X
ram(1) <= "00000000" & "001" & "10100";  -- Store in M(20)
ram(2) <= "00000000" & "100" & "00000";  -- IN A Y
ram(3) <= "00000000" & "100" & "10101";  -- Store in M(21)
ram(4) <= "00000000" & "101" & "10001";  -- JZ 17
ram(5) <= "00000000" & "000" & "10110";  -- Load M(22)
ram(6) <= "00000000" & "010" & "10100";  -- Add Acc + M(20)
ram(7) <= "00000000" & "001" & "10110";  -- Store in M(22)
ram(8) <= "00000000" & "000" & "10101";  -- Load M(21)
ram(9) <= "00000000" & "011" & "10111";  -- Sub Acc- M(23)
ram(10) <= "00000000" & "111" & "00011"; -- Jum ram(3)
```

```
ram(17) <= "00000000" & "000" & "10110"; -- Load M(22)
ram(18) <= "00000000" & "100" & "00001";  -- Out
ram(20) <= "00000000" & "000" & "00000";  -- M(20) = 0
ram(21) <= "00000000" & "000" & "00000";  -- M(21) = 0
ram(22) <= "00000000" & "000" & "00000";  -- M(22) = 0
ram(23) <= "00000000" & "000" & "00001";  -- M(23) = 1
```