MOS 6502 / MOS 6510

Deep Dive Bit Manipulation

ASL ROL SBC

ASL Shift Left One Bit (Memory or Accumulator)

```
C <- ←76543210→ <- 0              N Z C I D V
      <<<<<<<<                    + + + - - -
```

addressing	assembler	opc	bytes	cycls
accumulator	ASL A	0A	1	2
zeropage	ASL oper	06	2	5
zeropage,X	ASL oper,X	16	2	6
absolute	ASL oper	0E	3	6
absolute,X	ASL oper,X	1E	3	7

* add 1 to cycles if page boundary is crossed

** add 1 to cycles if branch occurs on same page
   add 2 to cycles if branch occurs to different page

Legend to Flags: + ... modified
                  - .... not modified
                  1 .... set
                  0 .... cleared
                 M6 .... memory bit 6
                 M7 .... memory bit 7

ROL Rotate One Bit Left (Memory or Accumulator)

```
C <- ←76543210→ <- C              N Z C I D V
      <<<<<<<<                    + + + - - -
```

addressing	assembler	opc	bytes	cycls
accumulator	ROL A	2A	1	2
zeropage	ROL oper	26	2	5
zeropage,X	ROL oper,X	36	2	6
absolute	ROL oper	2E	3	6
absolute,X	ROL oper,X	3E	3	7

* add 1 to cycles if page boundary is crossed

** add 1 to cycles if branch occurs on same page
   add 2 to cycles if branch occurs to different page

Legend to Flags: + ... modified
                  - .... not modified
                  1 .... set
                  0 .... cleared
                 M6 .... memory bit 6
                 M7 .... memory bit 7

SBC Subtract Memory from Accumulator with Borrow

A - M - C -> A

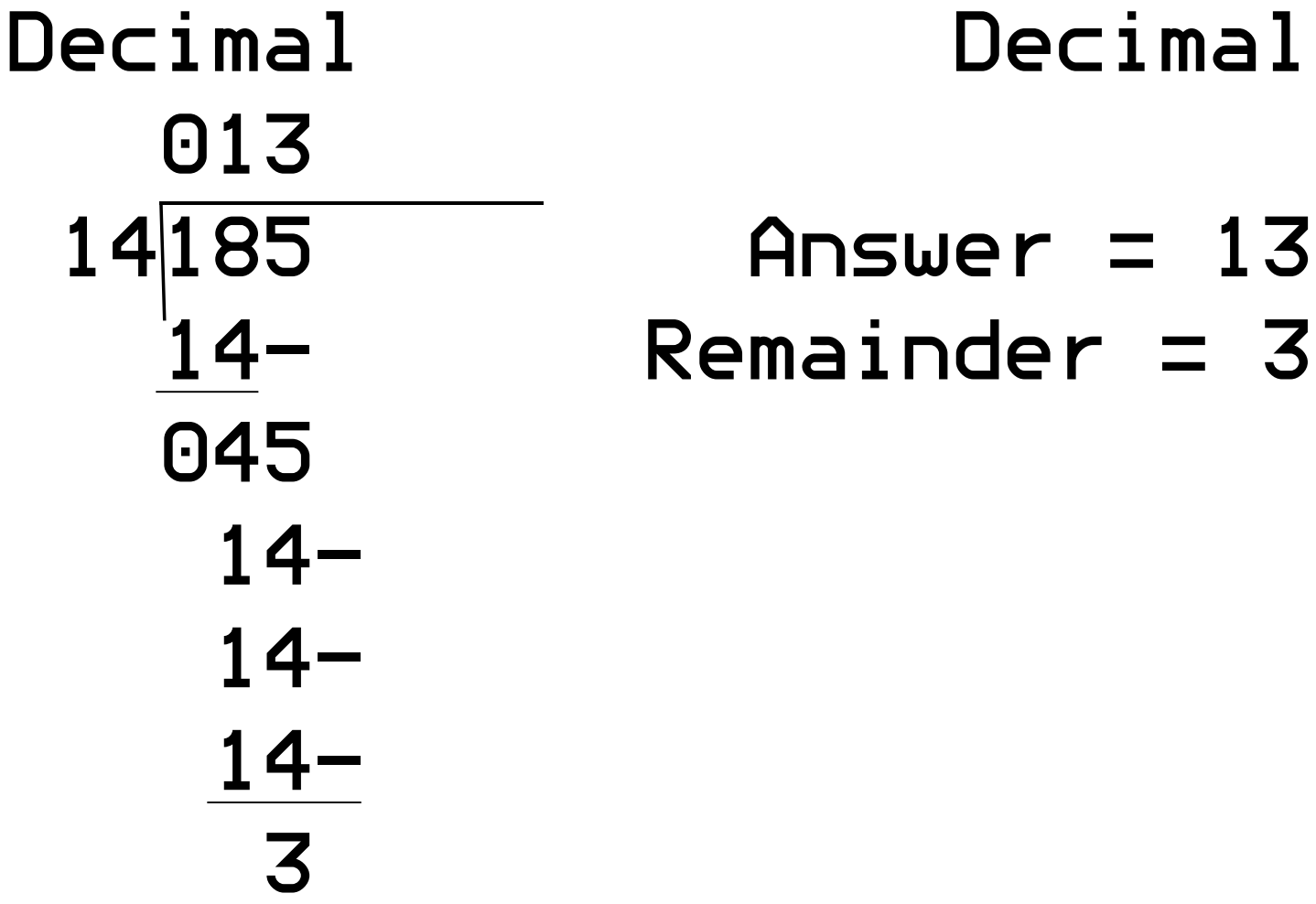
N	Z	C	I	D	V
+	+	+	-	-	+

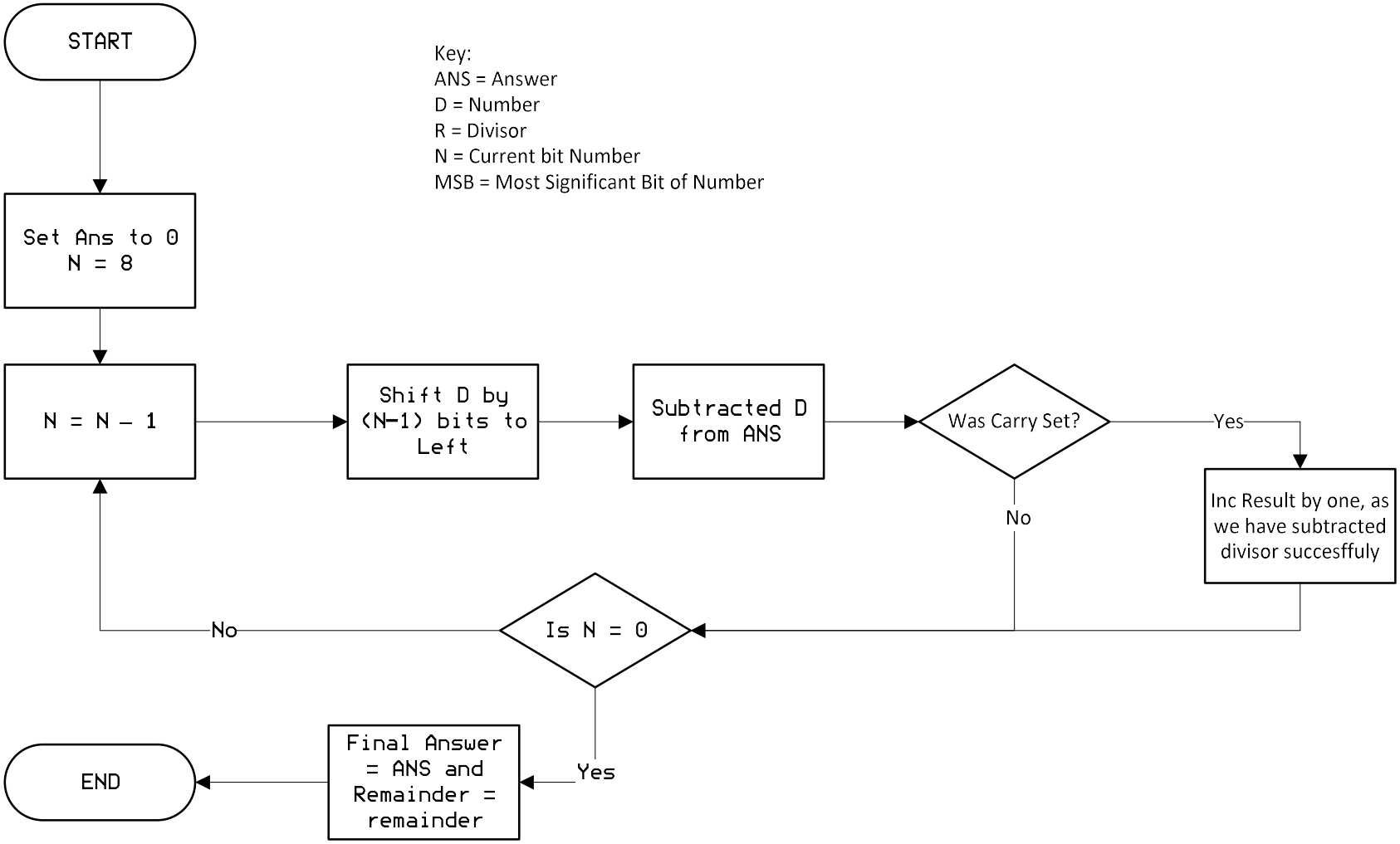
addressing	assembler	opc	bytes	cycls
immediate	SBC #oper	E9	2	2
zeropage	SBC oper	E5	2	3
zeropage,X	SBC oper,X	F5	2	4
absolute	SBC oper	ED	3	4
absolute,X	SBC oper,X	FD	3	4*
absolute,Y	SBC oper,Y	F9	3	4*
(indirect,X)	SBC (oper,X)	E1	2	6
(indirect),Y	SBC (oper),Y	F1	2	5*

* add 1 to cycles if page boundary is crossed

** add 1 to cycles if branch occurs on same page
   add 2 to cycles if branch occurs to different page

Legend to Flags: + ... modified
                  - .... not modified
                  1 .... set
                  0 .... cleared
                 M6 .... memory bit 6
                 M7 .... memory bit 7


Decimal
013
14|185
14-
045
14-
14-
14-
3
Decimal
Answer = 13
Remainder = 3


START
Key:
ANS = Answer
D = Number
R = Divisor
N = Current bit Number
MSB = Most Significant Bit of Number
Set Ans to 0
N = 8
N = N – 1
Shift D by (N-1) bits to Left
Subtracted D from ANS
Was Carry Set?
Yes
No
Inc Result by one, as we have subtracted divisor succesffuly
Is N = 0
No
Yes
Final Answer = ANS and Remainder = remainder
END