

```
; Unsigned Integer Division Routines (rev 2)
; by Omegamatrix
;
; Rev 1 (June 14, 2014)
; Divide by 6,10,12,20,24,26, and 28 have all been replace with new and
better routines.
;
; Rev 2 (June 21, 2014)
; Divide by 22 routines has been upgraded to one that saves 3 cycles, same
amount of bytes as before.
;
;
;
; To use these routines begin with unsigned value to be divided (0-255) in
the accumulator,
; and the routine will finish with the integer result in the accumulator.
;
; - All divisions (2-32) are covered below
; - X, Y, and BCD mode are not used by any of these routines
; - All these routines are constant cycles
; - Most routines require 1 temp register
```

```
;Divide by 2 (trival)
;1 byte, 2 cycles
    lsr
```

```
;Divide by 3
;18 bytes, 30 cycles
    sta temp
    lsr
    adc #21
    lsr
    adc temp
    ror
    lsr
    adc temp
    ror
    lsr
    adc temp
    ror
    lsr
```

```
;Divide by 4 (trival)
;2 bytes, 4 cycles
    lsr
```

```
    lsr

;Divide by 5
;18 bytes, 30 cycles
    sta temp
    lsr
    adc #13
    adc temp
    ror
    lsr
    lsr
    adc temp
    ror
    adc temp
    ror
    lsr
    lsr

;Divide by 6
;17 bytes, 30 cycles
    lsr
    sta temp
    lsr
    lsr
    adc temp
    ror
    lsr
    adc temp
    ror
    lsr
    adc temp
    ror
    lsr

;Divide by 7 (From December '84 Apple Assembly Line)
;15 bytes, 27 cycles
    sta temp
    lsr
    lsr
    lsr
    adc temp
    ror
    lsr
    lsr
    adc temp
    ror
    lsr
```

```
    lsr

;Divide by 8 (trival)
;3 bytes, 6 cycles
    lsr
    lsr
    lsr

;Divide by 9
;17 bytes, 30 cycles
    sta temp
    lsr
    lsr
    lsr
    adc temp
    ror
    adc temp
    ror
    adc temp
    ror
    lsr
    lsr
    lsr

;Divide by 10
;17 bytes, 30 cycles
    lsr
    sta temp
    lsr
    adc temp
    ror
    lsr
    lsr
    adc temp
    ror
    adc temp
    ror
    lsr
    lsr

;Divide by 11
;20 bytes, 35 cycles
    sta temp
    lsr
    lsr
    adc temp
    ror
```

```
adc temp
ror
adc temp
ror
lsr
adc temp
ror
lsr
lsr
lsr
```

```
;Divide by 12
;17 bytes, 30 cycles
```

```
lsr
lsr
sta temp
lsr
adc temp
ror
lsr
adc temp
ror
lsr
adc temp
ror
lsr
```

```
; Divide by 13
; 21 bytes, 37 cycles
```

```
sta temp
lsr
adc temp
ror
adc temp
ror
adc temp
ror
lsr
lsr
clc
adc temp
ror
lsr
lsr
lsr
```

```
;Divide by 14
```

```
;1/14 = 1/7 * 1/2
;16 bytes, 29 cycles
  sta temp
  lsr
  lsr
  lsr
  adc temp
  ror
  lsr
  lsr
  adc temp
  ror
  lsr
  lsr
  lsr

;Divide by 15
;14 bytes, 24 cycles
  sta temp
  lsr
  adc #4
  lsr
  lsr
  lsr
  adc temp
  ror
  lsr
  lsr
  lsr

;Divide by 16 (trival)
;4 bytes, 8 cycles
  lsr
  lsr
  lsr
  lsr

;Divide by 17
;18 bytes, 30 cycles
  sta temp
  lsr
  adc temp
  ror
  adc temp
  ror
  adc temp
  ror
  adc #0
```

```
lsr  
lsr  
lsr  
lsr
```

```
;Divide by 18 = 1/9 * 1/2  
;18 bytes, 32 cycles
```

```
sta temp  
lsr  
lsr  
lsr  
adc temp  
ror  
adc temp  
ror  
adc temp  
ror  
lsr  
lsr  
lsr  
lsr
```

```
;Divide by 19  
;17 bytes, 30 cycles
```

```
sta temp  
lsr  
adc temp  
ror  
lsr  
adc temp  
ror  
adc temp  
ror  
lsr  
lsr  
lsr  
lsr
```

```
;Divide by 20  
;18 bytes, 32 cycles
```

```
lsr  
lsr  
sta temp  
lsr  
adc temp  
ror  
lsr
```

```
lsr
adc temp
ror
adc temp
ror
lsr
lsr
```

```
;Divide by 21
;20 bytes, 36 cycles
```

```
sta temp
lsr
adc temp
ror
lsr
lsr
lsr
lsr
adc temp
ror
adc temp
ror
lsr
lsr
lsr
lsr
```

```
;Divide by 22
;21 bytes, 34 cycles
```

```
lsr
cmp #33
adc #0
sta temp
lsr
adc temp
ror
adc temp
ror
lsr
adc temp
ror
lsr
lsr
lsr
```

```
;Divide by 23
;19 bytes, 34 cycles
```

```
sta temp
```

```
lsr
lsr
lsr
adc temp
ror
adc temp
ror
lsr
adc temp
ror
lsr
lsr
lsr
lsr
```

```
;Divide by 24
;15 bytes, 27 cycles
```

```
lsr
lsr
lsr
sta temp
lsr
lsr
adc temp
ror
lsr
adc temp
ror
lsr
```

```
;Divide by 25
;16 bytes, 29 cycles
```

```
sta temp
lsr
lsr
lsr
adc temp
ror
lsr
adc temp
ror
lsr
lsr
lsr
lsr
```

```
;Divide by 26
```



```
;21 bytes, 37 cycles
```

```
lsr
sta temp
lsr
adc temp
ror
adc temp
ror
adc temp
ror
lsr
lsr
adc temp
ror
lsr
lsr
lsr
```

```
;Divide by 27
```

```
;15 bytes, 27 cycles
```

```
sta temp
lsr
adc temp
ror
lsr
lsr
adc temp
ror
lsr
lsr
lsr
lsr
```

```
;Divide by 28
```

```
;14 bytes, 24 cycles
```

```
lsr
lsr
sta temp
lsr
adc #2
lsr
lsr
adc temp
ror
lsr
lsr
```

```
;Divide by 29
```

```
;20 bytes, 36 cycles
```

```
  sta temp  
  lsr  
  lsr  
  adc temp  
  ror  
  adc temp  
  ror  
  lsr  
  lsr  
  lsr  
  adc temp  
  ror  
  lsr  
  lsr  
  lsr
```

```
;Divide by 30
```

```
;14 bytes, 26 cycles
```

```
  sta temp  
  lsr  
  lsr  
  lsr  
  lsr  
  sec  
  adc temp  
  ror  
  lsr  
  lsr  
  lsr  
  lsr
```

```
;Divide by 31
```

```
;14 bytes, 26 cycles
```

```
  sta temp  
  lsr  
  lsr  
  lsr  
  lsr  
  lsr  
  adc temp  
  ror  
  lsr  
  lsr  
  lsr  
  lsr
```

```
;Divide by 32 (trival)
;5 bytes, 10 cycles
    lsr
    lsr
    lsr
    lsr
    lsr
```

ref <https://atariage.com/forums/blogs/entry/10805-unsigned-integer-division-routines/>

From:

<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:

https://codebase64.org/doku.php?id=base:8bit_divide_by_constant_8bit_result

Last update: **2020-02-09 12:30**

