Z80 Routines:Math:Advance Math

From WikiTI

Contents

Introduction

These are routines designed for math of a slightly higher level. These don't necessarily contribute to everyday coding, but might be useful for an OS that handles such math (or programming language).

nCr Algorithm

This computes "n choose r" using an algorithm that makes use of both shadow registers and other calls. This can very likely be optimised, so feel free to edit with a new version.

```
;=============================================================
nCrHL_DE:
;=============================================================
;Inputs:
;     hl is "n"
;     de is "r"
;Outputs:
;     interrupts off
;     a is 0
;     bc is an intermediate result
;     de is "n"
;     hl is the result
;     a' is not changed
;     bc' is "r"+1
;     de' is the same as bc
;     hl' is "r" or the compliment, whichever is smaller
;=============================================================
     or a                      ;reset carry flag
     sbc hl,de
     ret c                     ;r should not be bigger than n
     sbc hl,de \ add hl,de
     jr nc,$+3
       ex de,hl
                               ;hl is R
     push de
     ld bc,1                   ;A
     exx
     pop de                    ;N
     ld bc,1                   ;C
     ld h,b \ ld l,c           ;D
nCrLoop:
     push de
     push hl
     call DE_Times_BC          ;Returns BC unchanged, DEHL is the product
     push hl \ exx \ pop de
     push hl
     call DE_Div_BC            ;Returns HL is the quotient, BC is not changed
```

```
    pop de
    push hl \ ex de,hl \ exx \ pop hl
    ld b,h \ ld c,l
    pop de \ add hl,de
    pop de \ inc de
    exx
    inc bc
    or a \ sbc hl,bc \ add hl,bc
    exx
    jr nc,nCrLoop
    ret
```

GCDHL_BC

This finds the greatest common divisor (GCD) of HL and BC.

```
GCDHL_BC:
;Inputs:
;     HL is a number
;     BC is a number
;Outputs:
;     A is 0
;     BC is the GCD
;     DE is 0
;Destroys:
;     HL
;Size:  25 bytes
;Speed: 30 to 49708 cycles
;       -As slow as about 126 times per second at 6MHz
;       -As fast as about 209715 times per second at 6MHz
;Speed break down:
;     If HL=BC, 30 cycles
;     24+1552x
;     If BC>HL, add 20 cycles
;     *x is from 1 to at most 32 (because we use 2 16-bit numbers)
;
    or a \ sbc hl,bc     ;B7ED42    19
    ret z                ;C8        5|11
    add hl,bc            ;09        11
    jr nc,$+8            ;3006      11|31
      ld a,h             ;7C        --
      ld h,b             ;60        --
      ld b,a             ;47        --
      ld a,l             ;7D        --
      ld l,c             ;69        --
      ld c,a             ;4F        --
Loop:
    call HL_Div_BC       ;CD****    1511    returns BC unchanged, DE is the remainder
    ld a,d \ or e        ;7AB2      8
    ret z                ;C8        5|11
    ld h,b \ ld l,c      ;6069      8
    ld b,d \ ld c,e      ;424B      8
    jr $-10              ;18F8      12
```

LCM

This is as simple as multiplying the two numbers and dividing by the GCD.

Credits and Contributions

- **Zeda (Xeda) Elnara** for the nCr and GCD algorithm

Retrieved from "http://wikiti.brandonw.net/index.php?title=Z80_Routines:Math:Advance_Math&oldid=9869"
Categories: Z80 Routines:Math | Z80 Routines

- This page was last modified on 9 January 2012, at 09:26.