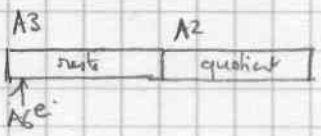
30 division de $|[A0]|$ par $|[A1]|$

sont [A2] et [A3] : $|[A0]| = [A2]^S * |[A1]| + [A3]$




```
XDIV1: CMP  #$4000,(A1)
       BEQ  ERRDV          → divis. par zéro
       MOVE.L  A0,A4
       MOVE.L  A1,A5
       BSR  XCMP1          compare |[A1]| et |[A0]|
       MOVE.L  A4,A0
       BCS  KA92           si |[a0]| < |[a1]|  mettre |[A0]| ; 0
       BEQ  KA93           si |[A0]| = |[A1]|  mettre 0 ; 1
                           ↓ |[A0]| > |[A1]|
       MOVE.L  A5,A1
```

entrée pour pgcd →

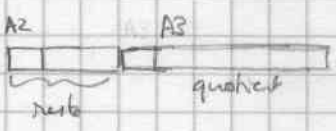
← ici il faut A0=A4 et A1=A5

```
XDIV1A: MOVE  (A1)+,d1
       AND.L  #$5FFF,d1     ⊗ .L
       BCLR  #14,d1
       BNE  GD74            → a1 court
       CMP  #4,d1
       BGT  GD81            → a1 long
       BNE  \1
       MOVE.L  (A1),D1
       BRA  GD74

   \1: MOVE  (A1),D1
GD74: CMP.L  #1,d1          divise |[a0]| par d1.L
       BEQ  KA94            si d1=1  mettre 0 ; |[A0]|
       MOVEQ  #0,d0
       MOVE  (A0)+,d0
       AND.L  #$5FFF,d0
       BCLR  #14,d0
       BNE  \2              a0 court
       CMP  #4,d0
       BGT  GD77            → a0 long
       BNE  \1
       MOVE.L  (A0),d0
       BRA  \2
```

```
\1: MOVE (A0),d0
\2: DIVUL.L d1,d2,d0        divise d0 32 par d1 32 → d2.L = r ; d0.L = q
    EXG D0,D2
    BSR XPOSEL              pose d0 = reste
    MOVE.L A2,A3
    MOVE.L D2,D0
    BRA XPOSEL              pose quotient

GD77: MOVE.L A6,A2          divise [A0] long par D1.L → D0
    LEA 8(A2),A3
    LEA (A3,D0),A6
    BSR VERA6               vérif
    MOVE.L A3,-(SP)
    MOVEQ #0,D2
    SUBQ #2,d0
    ASR #2,d0
    BCS \2
    MOVEQ #0,D3
    MOVE (A0)+,D3
    BRA \3
\2: MOVE.L (A0)+,D3
\3: DIVU.L D1,D2,D3         reste d2
                            quotient d3
    MOVE.L D3,(A3)+
    DBRA D0,\2
    MOVE.L A2,A6
    MOVE.L D2,D0
    BSR XPOSEL              met reste
    MOVE.L A6,A0
    MOVE.L A3,A6
    MOVE.L (SP)+,A2         ← MOVE.L A2,A3 ⊗
(répète GD66) → \4: TST (A2)+
    BEQ \4
    SUBQ #2,A2
    MOVE.L A6,D0
    SUB.L A2,D0
    MOVE D0,-(A2)           ← SUBQ #2,D0 ; BNE \5 ; BSR KA38   rectifie si q < $2000
\5: MOVE.L A0,-(SP)
    BSR XLB76
    MOVE.L (SP)+,A2
    RTS
```



2 → 0 → 0
4 → 2 → 0 ⊡
6 → 4 → 1
8 → 6 → 1 ⊡

30 divise {A4}A0 (D0 = a) par {A5} A1 (D1 = b) ← description / ← longueurs

GD81: MOVE (A0),D0 — {ADDQ #2,D0 / ASR #2,D0} — D1 Normalise
AND #$1FFF,D0 — m+n
BFFFO (A1),0,16,D2 — $0 \le d2 \le 15$ (car (A1).W ≠ 0)
BTST #1,D1
BEQ \1 ⊗ — si d1 n'est pas multiple de 4 ajoute 16 à d2
ADD #16,D2

facteur de normalisat. $= 2^{d2} = 2^{\psi}$

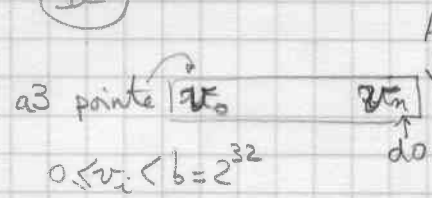
\1: MOVEM.L D7/A6,-(SP) ← place pour v_0
CLR (A6)+
MOVE.L A6,A3 — MOVEM D0/D2,-(SP)
MOVE.L A5,A0 — ① 91 pose $v = 2^{\psi} b = \{A3\}$
BSR XROT ← MOVE.L A6,A1 ⊗
MOVE.L A4,A0
ADDQ #4,A6 (pour mettre 0 devant u)
BSR XROT — ① 91 pose $u = 2^{\psi} a = \{A2\} = \{A1\}$

MOVE (A3),d0 ← AND #$1FFF,d0 ⊗ — n
ASR #2,d0
MOVE (A2)+,d2
AND #$1FFF,d2
ASR #2,d2
BCC \2 ← ST TT;*
CLR -(A2)
ADDQ #1,d2
\2: CLR (A3)+ — $v_0 = 0$ (partie basse)
MOVEM.L (A3),D5/D6 — $d5 = v_1$, $d6 = v_2$

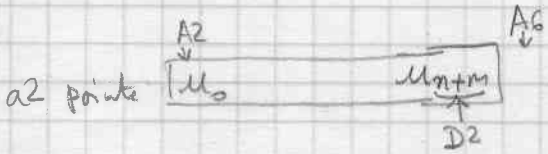
\2: CMP (SP),D2
BNE \3
ST TT+1;*
CLR.L -(A2)
\3: MOVE (SP)+,D2 — n+m

↓ si impair multiple de 4

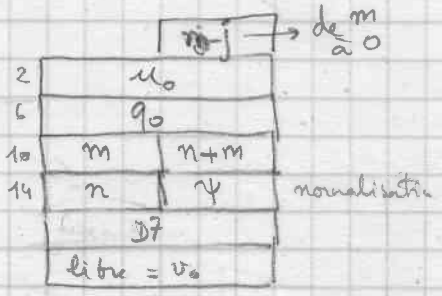
D2



$0 \le v_i < b = 2^{32}$



MOVE D0,-(SP) — n
MOVE D2,-(SP) — n+m
SUB D0,D2
MOVE D2,-(SP) — m ← CLR.L (A6)+
MOVEM.L A2/A6,-(SP)
MOVE D2,-(SP)
ASL #2,D2
ADD D2,A6 — vérif mémoire
BSR VERA6
SUB D2,A6



```
GE73:MOVEQ #-1,D4        d4 = b-1
     CMP.L (A2),D5
     BEQ \2
                         d2:d4 = u_j : u_{j+1}
     MOVEM.L (A2),D2/D4
     DIVU.L D5,D2,D4     d2 = r , d4 = q̂
     ST TT+2;*t
     BRA \3
\2:  MOVE.L 4(A2),D2     si u_j = v_1   q̂ = b-1   r = v_1 + u_{j+1}
     ADD.L D6,D2    <- ST TT+3;*t
     BCS GE77            -> r > b
     ST TT+4;*t

\3:  MOVE.L 8(A2),D3     d2:d3 = [r]:[u_{j+2}] = (u_j b + u_{j+1} - q̂ v_1) b + u_{j+2}
     MOVE.L D6,D1
     MULU.L D4,D0,D1     } d0:d1 = q̂ . v_2
     SUB.L D1,D3         d2:d3 = [r]:[u_{j+2}] - q̂ v_2
     SUBX.L D0,D2
     BCC GE77  <- ST TT+5;*t   -> u_j : u_{j+1} : u_{j+2} >= q̂ (v_1 : v_2)
     MOVEQ #0,D0 (X)
\4:  SUBQ.L #1,D4        q̂ -> q̂ - 1
                         nouveau d2:d3 = d2:d3 + v_1 : v_2
     ADD.L D6,D3
     ADDX.L D5,D2 (X)
     BCC \4
```

(D3)
A2 sur u_j
A1 après v_n
A6 en libre pour q_i
D5 = v_1
D6 = v_2

```
GE77: MOVE  14(SP), d2          (n)
⊗     LEA   4(A2, d2.W*4), A3   A3 après u_{j+n}
      MOVE.L A1, A0             A0 après v_n
      MOVEQ #0, D0              d3:d0 ≤ b ))
      MOVEQ #0, D3
\1:   MOVE.L -(A0), D1
      MULU.L D4, D7:D1
      ADD.L  D0, D1
      ADDX.L D3, D7                          α
      MOVEQ #0, D3
      MOVE.L D7, D0
      SUB.L  D1, -(A3)
      BCC   \2
      ST    TT+6 ;*t
      ADDQ.L #1, D0
      ADDX.L D3, D3   ← TST.L D3 ⊗
      BEQ   \2 ;*t
      ST    TT+6 ;*t
\2:   DBRA  D2, \1
      OR.L  D3, D0
      BEQ   \4
      ST    TT+6 ;*t
      SUBQ.L #1, D4   )met aussi x=0
repiteGE77  MOVE  14(SP), d2   (n)
      LEA   4(A2, d2.W*4), A3   ⊗
      MOVE.L A1, A0
\3:   ADDX.L -(A0), -(A3)
      DBRA  D2, \3
\4:   MOVE.L D4, (A6)+          qj
      MOVE  (SP)+, D0           j
      SUBQ #1, D0
      BMI   GE82                → fin
      MOVE  D0, -(SP)
      ADDQ  #4, A2
      BRA   GE73
```

D4 soustraire à $u_j, \dots u_{j+n}$ $\hat{q}(v_1 \dots v_n)$

inutile ? (mais ne perd rien en temps)
on a plutôt $d3:d0 \leq b-1$ en ⊗ \1
$d7:d1 \leq (b-1)^2 + b-1 = b^2-b$ en α
sub.l d1,-(a3) donne une retenue que si $d7:d1 < b^2-b$; addq #1,d0 ne donne pas de retenue.

D5 carry ?

$\hat{q}$ trop grand D6 add back

D7 $j = j+1$

```
GE82: MOVE.L  (SP)+, A1   <-- MOVE.L 12(SP),D7   (en cas d'erreur mémoire)
      MOVE    6(SP), D1   m+n
      ADD     D1,D1
      ADDQ    #1,D1       2(m+n)+1   (nb de mots - 1)
      BSR     XPOSEA1     (1)53   pose r2^Ψ en libre=A2
      MOVE.L  A2,A0
      MOVE    10(SP), D2  Ψ
      MOVE.L  16(SP),A6
      NEG     D2
      BSR     XROT        r en A6^e
      MOVE.L  (SP)+, A1   q0
      MOVE    (SP)+, D1   m
      ADD     D1,D1
      ADDQ    #1,D1
      BSR     XPOSEA1     q en A2
      ADD     #10, SP
      MOVE.L  (SP)+,A3    r en A3
      RTS
```

KA92, KA93, KA94 } commun avec XDIV1/ST