

TP : Réalisation d'une machine algorithmique

5.3 Modification de l'algorithme

1.

```
r = a
while (r ≥ n) {
    r = r - n;
}
```

Pour calculer le modulo d'un entier en n'utilisant que des soustractions, l'idée est de retirer répétitivement la valeur n de l'entier a jusqu'à ce que le résultat soit strictement inférieur à n. Autrement dit, si l'on souhaite calculer $r = a \bmod n$, on peut procéder de la manière suivante :

Cette méthode garantit que le résultat final r satisfait $0 \leq r < n$.

2.

```
#include <stdio.h>
#include <stdlib.h>

void main(int argc, char * argv[])
{
    unsigned long int m, exp, n;
    unsigned long int code, temp;

    if (argc != 4) exit(0);
    sscanf(argv[1], "%ld", &m);
    sscanf(argv[2], "%ld", &exp);
    sscanf(argv[3], "%ld", &n);

    code = 1;
    while (exp > 0) // exp != 0 donc flag (not Z)
    {
        if ((exp & 1) > 0) //(LSB == 1) Last Significant Bit
        {
            // code = (code * m) % n, calculé par soustractions répétées
            code = code * m;
            while (code >= n) //while C
            {
                code = code - n;
            }
        }
        exp = exp >> 1; // maj flag Z
        // m = (m * m) % n, calculé par soustractions répétées
        m = m * m;
        while (m >= n) //while C
        {
            m = m - n;
        }
    }
    printf("Resultat chiffrement %ld\n", code);
}
```

2.

3.

```
-- Définition d'un soustracteur n bits
include "../multv6base2.lus"
include "../entier6.lus"
include "../UALCryptoV6.lus"

function subnbits<<const n: int>>(A, B: bool^n) returns (S: bool^n; Borrow: bool);
var
  B_out: bool^n;
let
  -- Complément à 2 de B pour la soustraction (A - B = A + (~B + 1))
  B_out = map<<not, n>>(B); -- Complément à 1 en inversant les bits
  (S, Borrow) = add_n_bits<<n>>(A, B_out, true); -- +1 pour le complément à 2
tel;

-- Notre définition de l'UAL
function UAL<<const n: int>>(
  OpUAL: bool^2;      -- Commande de l'UAL (2 bits)
  Op1, Op2: bool^n;   -- Opérandes (n bits)
)
returns (
  C, Z, Odd: bool;    -- Drapeaux de statut
  Result: bool^n      -- Résultat de l'UAL (n bits)
);
var
  Mult_Result: bool^(2*n); -- Résultat de la multiplication (2*n bits)
  Sub_Result: bool^n;      -- Résultat de la soustraction
  Borrow: bool;            -- Retenue de la soustraction
  Shift_Result: bool^n;    -- Résultat du décalage
  Exp_Is_Odd: bool;        -- EXP & 1 = 1 ?
let
  -- Multiplication (code * m)
  -- On évite de faire le calcul selon OpUAL
  Mult_Result = multnbits<<n>>(Op1, Op2);
  -- Soustraction (pour le modulo)
  (Sub_Result, Borrow) = subnbits<<n>>(Op1, Op2);
  -- Décalage à droite (exp >> 1)
  Shift_Result = shiftl<<n>>(Op1, false) ;

  -- Résultat final (selon OpUAL)
  Result = map <<mux4, n>>(
    OpUAL[1]^n, OpUAL[0]^n,
    Sub_Result, Mult_Result[0..n-1], Shift_Result, Op1
  );
  -- Calcul des flags
  Z = (Result = false^n); -- Z: EXP = 0 ?
  Exp_Is_Odd = Result[0]; -- EXP & 1 = 1 ?

  C = not Borrow;        -- Emprunt de la soustraction
  Odd = Exp_Is_Odd;      -- EXP & 1 = 1 ?
```

```
tel

-- Exemple d'utilisation de l'UAL comme l'exemple donné dans le UALcryptoV6
function ualaff_2<<const n:int>>( A , B : bool^n ; OpUAL : bool^2 )
returns ( C,Z,Odd : bool ; T : bool^n; AE,BE,R:int );
let
(C,Z,Odd,T)=UAL<<n>>( OpUAL, A , B );

R=entiernat<<n>>(T);
AE=entiernat<<n>>(A);
BE=entiernat<<n>>(B);

tel;
node ualinst_2=ualaff<<a>>;
```

4.

```
-- Version sans CK
-----
include "../memoire.lus"

node RegN <<const n : int>>
(ent : bool^n; -- bus de données à charger
char : bool; -- signal de chargement commun
reset : bool; -- reset synchrone (met tous les bits à 0)
set : bool) -- set asynchrone (met tous les bits à 1)
returns
(sort : bool^n); -- bus de sortie (état du registre)
let
-- map applique n fois le noeud bascule,
-- en dupliquant char, reset et set en vecteurs de taille n
sort = map <<bascule; n>>
( ent, -- ent[i]
char^n, -- char pour chaque bit
reset^n, -- reset pour chaque bit
set^n -- set pour chaque bit
);
tel
-- Pour juste 4 bits on instancie :
node Reg4 = RegN<<4>>;

-- Version avec CK
-----

-- Registre n bits avec chargement conditionnel
node registre_n_bits<<const n: int>>{
CK: bool; -- Horloge globale du circuit --On n'utilise pas dans le code juste pour la clarté
chM: bool; -- Signal de chargement (connecté à CHAR des bascules)
reset: bool; -- Reset global (actif à 1)
set: bool; -- Set global (actif à 1)
D: bool^n -- Donnée d'entrée
}
returns (
Q: bool^n -- Sortie du registre
);
let
-- Génération des n bascules en parallèle
Q = map<<bascule; n>>(D, chM^n, reset^n, set^n);

-- Connexion implicite de l'horloge :
-- Le signal CK est implicite dans l'opérateur -> et pre de la bascule
tel;

--exemple
-- Registre M de 8 bits
node registre_M(
Clock: bool; -- Horloge --On n'utilise pas dans le code juste pour la clarté
load_M: bool; -- Commande de chargement
rst: bool; -- Reset
data_in: bool^8 -- Bus d'entrée
)
returns (
M_out: bool^8 -- Sortie
);
let
M_out = registre_n_bits<<8>>(Clock, load_M, rst, false, data_in);
-- set mis à false car on n'initialise pas le registre M avec 1
tel;
```

5.

```
-- 1) On inclut d'abord tout ce qu'il faut :
--      * Les bascules les registres et UAL à n bits
include "../memoire.lus"
include "3.lus"
include "4.lus"
include "../entierV6.lus"
include "../multV6base2.lus"
include "../UALCryptoV6.lus"

-----
-- 2) Node P0 générique à n bits -----
-----

node P0 <<const n:int>>()
  Clock      : bool;      -- horloge
  Reset      : bool;      -- reset synchrone =1
  selExt     : bool;      -- 1 = charger depuis busIn (initialisation), 0 = recirculer UAL
  busIn      : bool^n;    -- bus d'entrée externe (M, EXP ou N)
  chCode     : bool;      -- load CODE
  chM        : bool;      -- load M
  chExp      : bool;      -- load EXP
  chN        : bool;      -- load N
  selOp1     : bool^2;    -- code 00..11 pour choisir CODE/M/EXP/N en op1
  selOp2     : bool;      -- code 0 1 pour choisir M/N en op2
  opUAL      : bool^2;    -- commande UAL (00=sub, 01=mul, 10=shr1, 11=pass)
  mise_1     : bool      -- mise à 1 de CODE (pour le premier tour)
)

returns (
  busOut     : bool^n;    -- seule sortie visible : CODE
  Z, C, Odd  : bool;      -- flags Zero et Carry (emprunt)
  CodeInt, MInt, NInt, ExpInt : int -- pour le debug
);
var
  CODE, M, EXP, Nreg : bool^n; -- registres internes
  op1, op2           : bool^n; -- bus opérands vers l'UAL
  resUAL             : bool^n; -- résultat de l'UAL
  Borrow, Zflag      : bool;   -- emprunt, zero renvoyés par l'UAL
  setVecCode : bool^n; -- vecteur de set pour CODE
let
  -----
  setVecCode = [true] | false^(n-1); -- vecteur à 1 seulement en position 0
  -- CODE = if mise_1 then [true] | registre_n_bits<<n>>(Clock, chCode, Reset, false,
  --           if selExt then busIn else resUAL)
  --           else registre_n_bits<<n>>(Clock, chCode, Reset, false,
  --           if selExt then busIn else resUAL);

  -- Instanciation "à la main" de CODE via map<<bascule;n>> pour passer setVecCode
  CODE = map <<bascule; n>>()
    -- ent : si phase init (selExt) on prend busIn, sinon resUAL
    if selExt then busIn else resUAL,
    -- char : chCode répliqué
    chCode^n,
    -- reset : Reset répliqué
    Reset^n,
    -- set   : notre setVecCode
    if mise_1 then setVecCode else false^n
  );

  -- On charge CODE en premier, puis les autres registres
  --peut etre false au lieu de Reset and not mise_1
  M = registre_n_bits<<n>>(Clock, chM, Reset and not mise_1, false,
```

```
-- On charge CODE en premier, puis les autres registres
--peut etre false au lieu de Reset and not mise_1
M   = registre_n_bits<<n>>(Clock, chM, Reset and not mise_1, false,
    | | | | if selExt then busIn else resUAL);
EXP = registre_n_bits<<n>>(Clock, chExp, Reset and not mise_1, false,
    | | | | if selExt then busIn else resUAL);
Nreg = registre_n_bits<<n>>(Clock, chN, Reset and not mise_1, false,
    | | | | if selExt then busIn else resUAL);

-----
-- 4) Multiplexeurs 4:1 et 2:1 pour sélectionner les deux opérandes
op1 = map <<mux4; n>>(
    | | | | selOp1[1]^n, selOp1[0]^n,
    | | | | CODE, M, EXP, Nreg);

op2 = map <<mux1; n>>(
    | | | | selOp2^n,
    | | | | Nreg, M);

-----
-- 5) Appel de l'UAL (issue de 3.lus) :
--   UAL(opUAL, op1, op2) returns (C, Z, Odd, Result)
(Borrow, Zflag, Odd, resUAL) = UAL<<n>>(opUAL, op1, op2);

-----
-- 6) Bus de sortie et drapeaux
busOut = CODE;    -- on n'exporte que CODE
C       = not Borrow; -- emprunt de la soustraction
Z       = Zflag;   -- résultat nul ?

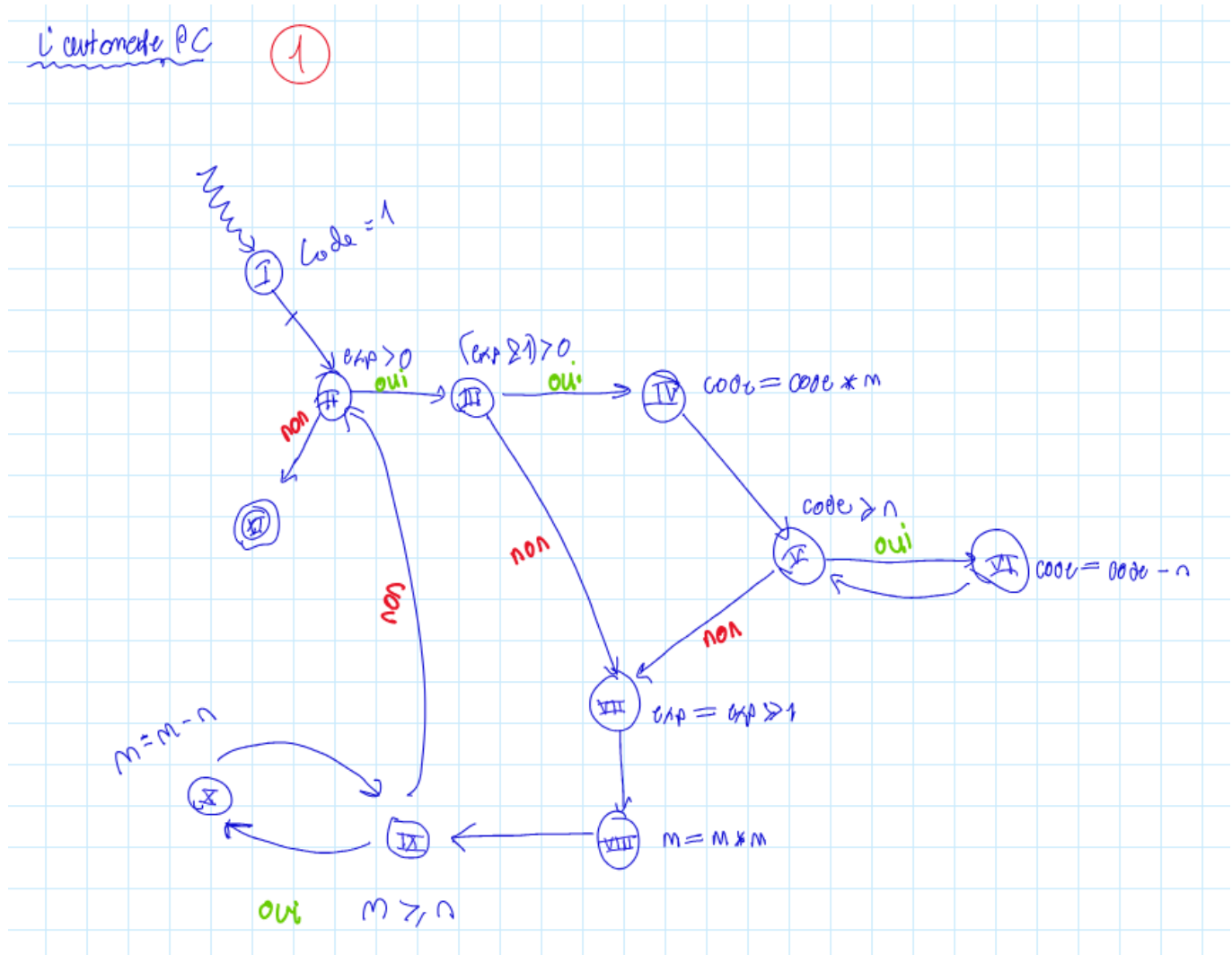
--pour le voir mieux --debug
CodeInt = entiernat<<n>>(CODE);
MInt    = entiernat<<n>>(M);
NInt    = entiernat<<n>>(Nreg);
ExpInt  = entiernat<<n>>(EXP);

tel

-- Instanciation pour n=8 bits, par exemple
node P08 = P0<<8>>;
```

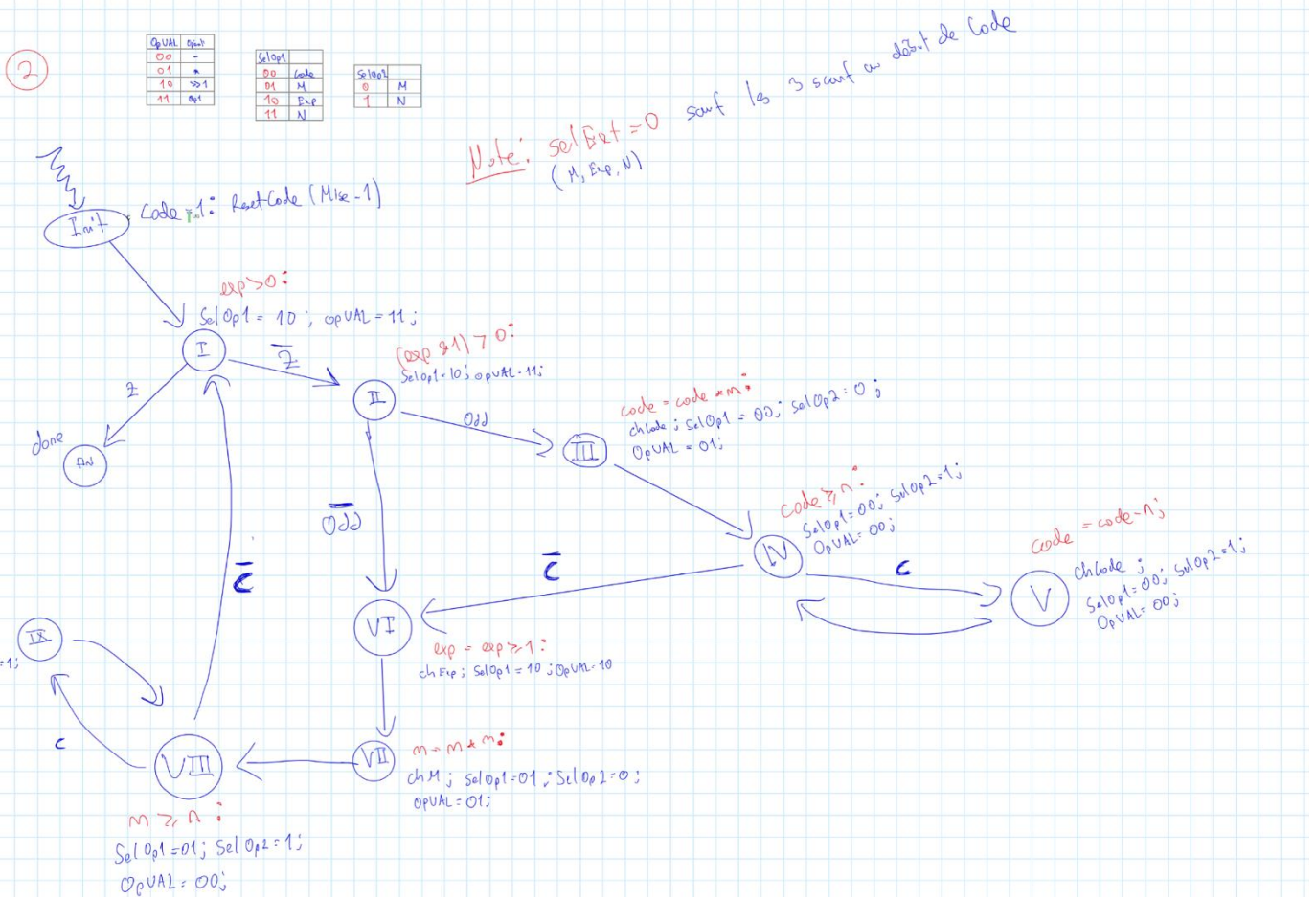
5.5 La partie contrôle

1.



2.

N.B : On n'indique pas explicitement les valeurs qui ne sont pas utilisées dans l'état en question.



3.

```
-----
include "../memoire.lus"

node PC(
  CLK   : bool;      -- Horloge --ce n'est pas utilisé dans le code , juste pour la simulation
  RESET : bool;      -- Reset synchrone (actif haut)
  Zexp  : bool;      -- Flag EXP = 0
  C     : bool;      -- Flag Carry (pour modulo)
  Odd   : bool;      -- Flag EXP impair (LSB)
)
returns (
  -- Signaux de chargement
  chExp, chCode, chN, chM : bool;
  -- Sélection entrée
  selExt : bool;
  -- Contrôle multiplexeurs
  selOp1 : bool^2;
  selOp2 : bool;
  -- Commande UAL
  opUAL : bool^2;
  -- Indicateur de fin
  done : bool;
  --signale de mise à 1
  mise_1 : bool;
  -- Signaux de débogage
  etat_courant : bool^11
);
var
  st, st_next : bool^11;  -- vecteurs un-among-11

let
  st[0] = bascule(st_next[0], true , false, RESET); --initial;
  st[1..10] = map <<bascule;10>>(
    st_next[1..10], -- Entrée D
    true^10,        -- CHAR toujours actif
    RESET^10,       -- Reset synchrone
    false^10        -- Set initial
  );
  -----
  -- 3) Machine à états (version optimisée)
  st_next = [
    -- État 0 (initCode) → État 1 (loopTest)
    RESET,

    -- État 1 : exp>0
    st[0] or (st[8] and not C),

    -- État 2 : (exp & 1) > 0
    st[1] and not Zexp ,

    -- État 3: code = code * m
    st[2] and Odd,

    -- État 4 : code >= n
    st[3] or st[5],

    -- État 5: cpde = code - n;
    st[4] and C,

    -- État 6: exp = exp >> 1
    (st[2] and not Odd) or (st[4] and not C),
  ]

```

```
-- État 6: exp = exp >> 1
(st[2] and not Odd) or (st[4] and not C),

-- État 7 : m = m*n
st[6],

-- État 8 : m>= n
st[7] or st[9],

-- État 9 : m = m - n
st[8] and C,

-- État 10 : m = m >> 1
st[1] and Zexp
];

-----
-- 4) Génération des signaux de contrôle (version clarifiée)
mise_1 = st[0];
-- Chargements initiaux
chCode = st[0] or st[3] or st[5] ;
chExp = st[6];
chM = st[7] or st[9];

-- On n'utilise pas selExt, chN comme nous ne faisons que les affectations au debut de trois scanf
selExt = st[0];
chN = false;

-- Sélection entrée externe pendant l'init
--selExt = st[0];

-- Contrôle des multiplexeurs (table de vérité optimisée)
selOp1 = [
  -- Bit 0: 1 pour M
  st[7] or st[8] or st[9],

  -- Bit 1: 1 pour Exp
  st[1] or st[2] or st[6]
];

-- selOp2 = 1 pour N dans multiply(5)/shift(6)/square(7)
selOp2 = st[4] or st[5] or st[8] or st[9];

-- Commande UAL (nouveau codage plus lisible)
opUAL = [
  st[1] or st[2] or st[3] or st[7],
  st[1] or st[2] or st[6]
];

-- Signal de fin
done = st[10];

-- Nouveau: visualisation de l'état courant pour débogage
etat_courant = st;
tel
```

L'initialisation du circuit repose sur un **reset synchrone actif-haut** partagé par la partie contrôle (PC) et la partie opérative (PO). Au premier front montant de l'horloge après assertion de **RESET=1**, la bascule st[0] reçoit sa ligne **set** et passe à 1, tandis que toutes les autres bascules st[1..10] sont simultanément remises à 0 grâce à leur ligne **reset**. Cet unique "1" dans le vecteur d'état identifie l'**état 0** de la machine de contrôle et, via le décodage de cet état, active automatiquement les signaux mise_1 = 1, selExt = 1, chCode = 1 (ainsi que, si nécessaire, chM, chExp et chN).

Dans la PO, **mise_1=1** alimente la ligne **set** du registre CODE avec un vecteur [1,0,...,0], ce qui lui donne la valeur 1, tandis que **selExt=1** et **chCode=1** ordonnent à tous les registres (CODE, M, EXP, N) de charger depuis le bus d'entrée leurs valeurs initiales. Dès que **RESET** retombe à 0, la machine conserve ce vecteur d'état "1 000...0" un cycle puis passe à l'état 1, lançant ainsi l'exécution normale de l'algorithme de chiffrement.

5. La réunion de deux parties PC et PO : top.lus

```
-- top.lus
-- dépendances : PC.lus, PO.lus (votre 5.lus renommé), multv6base2.lus, UALCryptoV6.lus
include "PC.lus"
include "../PO/5.lus"

node top_debug <<const n:int>>{
  begin      : bool;    -- checkbox "begin"
  chM        : bool;    -- checkbox "chM"
  chEXP      : bool;    -- checkbox "chEXP"
  chN        : bool;    -- checkbox "chN"
  input      : int;     -- slider "input" (valeur entière à charger)
}

returns (
  m      : int; -- affichage registre M
  exp    : int; -- affichage registre EXP
  nreg   : int; -- affichage registre N
  end_   : bool; -- flag "end_"
  q      : int; -- copie de "code" si vous voulez un label supplémentaire
  -- Signaux de débogage
  etat_courant : bool^11;

  Sig_chExp : bool;
  Sig_chCode : bool;
  Sig_chN : bool;
  Sig_chM : bool;

  mise_1 : bool;

  selOp1 : bool^2; -- commande UAL (issue de PC)
  selOp2 : bool;
  opUAL : bool^2; -- commande UAL (issue de PC)

  Sig_C : bool; -- flag Carry
  Sig_Z : bool; -- flag Zero
  Sig_Odd : bool; -- flag Odd
);

var
  Z, C, Odd : bool; -- flags retour UAL
  --selOp1 : bool^2; -- commande UAL (issue de PC)
  --selOp2 : bool;
  --opUAL : bool^2;
  -- vecteur de bits issu du slider
  busIn : bool^n;
  -- bus interne de sortie PO
  code_bits : bool^n;

  reset : bool; -- reset PO

  code : int; -- code de sortie
  --signale de mise à 1

  Sig_chM_interne : bool;
  Sig_chExp_interne : bool;
  Sig_chN_interne : bool;
  Sig_chCode_interne : bool;

  -- vecteurs de bits issus des registres
  selExt : bool; -- 1 = charger depuis busIn (initialisation), 0 = recirculer UAL
let
```

```
64 let
65 -----
66 -- conversion de l'entier 'input' en bus de n bits (LSB à l'indice 0)
67 busIn = natier16(input);
68 -----
69 -----
70 -- Partie opérative (P0) : on charge manuellement via 'begin', chM, chEXP, chN
71 ( code_bits, Z, C, Odd, code, m, nreg, exp ) =
72   PO<<n>>(false,
73     reset,      -- reset P0 = false (on ne reset pas ici)
74     not begin,  -- selExt = !begin
75     busIn,      -- bus d'entrée
76     Sig_chCode_interne, -- chCode = begin (on force reset à 1 pour initialiser CODE=1)
77     Sig_chM_interne,   -- chM   = checkbox chM
78     Sig_chExp_interne, -- chExp  = checkbox chEXP
79     Sig_chN_interne,   -- chN   = checkbox chN
80     selOp1,            -- commande alu issue de PC
81     selOp2,
82     opUAL,
83     mise_1
84   );
85
86   Sig_chExp_interne = mux1(begin, Sig_chExp, chEXP);
87   Sig_chN_interne = mux1(begin, Sig_chN, chN);
88   Sig_chM_interne = mux1(begin, Sig_chM, chM);
89   Sig_chCode_interne = mux1(begin, Sig_chCode, false);
90
91   reset = (not pre begin) and begin;
92
93 -----
94 -----
95 -- Partie contrôle (PC) : on reset PC quand begin=0-1, on récupère
96 -- uniquement les commandes sorties; on ignore ici selExt/chCode/ch*
97 ( Sig_chExp, Sig_chCode, Sig_chN, Sig_chM,
98   selExt, selOp1, selOp2,
99   opUAL, end_, mise_1, etat_courant ) =
100   PC(false,
101     not begin, -- RESET_PC = 1 quand begin passe à 0-1 (synchronous)
102     Z, C, Odd);
103 -----
104 -----
105 -- sorties finales
106 q = entiernat<<n>>(code_bits); -- alias si besoin
107 Sig_C = C;
108 Sig_Z = Z;
109 Sig_Odd = Odd;
110
111 tel;
112 -----
113 -----
114 -- utilisation_top_debug
115 -- inst_top16 lus
116
117 -- On fixe n = 16
118 node top16(
119   begin : bool;
120   chM : bool;
121   chEXP : bool;
122   chN : bool;
123   input : int
124 )
125 returns (
126   m : int;
```

```
123   input  : int
124   )
125   returns (
126   m      : int;
127   exp    : int;
128   n      : int;
129   end_    : bool;
130   q      : int;
131   -- Signaux de débogage
132   etat_courant : bool^11;
133   Sig_chExp : bool;
134   Sig_chCode : bool;
135   Sig_chN : bool;
136   Sig_chM : bool;
137   mise_1 : bool;
138   selOp1 : bool^2; -- commande UAL (issue de PC)
139   selOp2 : bool;
140   opUAL : bool^2; -- commande UAL (issue de PC)
141   Sig_C : bool; -- flag Carry
142   Sig_Z : bool; -- flag Zero
143   Sig_Odd : bool; -- flag Odd
144   );
145   let
146   -- On déplie top_debug avec n=8
147   (m, exp, n, end_, q, etat_courant, Sig_chExp, Sig_chCode, Sig_chN, Sig_chM, mise_1, selOp1, selOp2, opUAL, Sig_C, Sig_Z, Sig_Odd) =
148   top_debug<<16>>(begin, chM, chEXP, chN, input);
149   tel;
150
151
152
153   -- natier :
154
155   -- natier8.lus
156   -- Convertit un entier i en vecteur de 8 bits (LSB en position 0)
157   function natier8(i:int) returns (b: bool^8);
158   let
159   b[0] = (i mod 2) = 1;
160   b[1] = ((i / 2) mod 2) = 1;
161   b[2] = ((i / 4) mod 2) = 1;
162   b[3] = ((i / 8) mod 2) = 1;
163   b[4] = ((i / 16) mod 2) = 1;
164   b[5] = ((i / 32) mod 2) = 1;
165   b[6] = ((i / 64) mod 2) = 1;
166   b[7] = ((i / 128) mod 2) = 1;
167   tel;
168
169   -- Convertit un entier i en vecteur de 16 bits (LSB en position 0)
170   function natier16(i:int) returns (b: bool^16);
171   let
172   b[0] = (i mod 2) = 1;
173   b[1] = ((i / 2) mod 2) = 1;
174   b[2] = ((i / 4) mod 2) = 1;
175   b[3] = ((i / 8) mod 2) = 1;
176   b[4] = ((i / 16) mod 2) = 1;
177   b[5] = ((i / 32) mod 2) = 1;
178   b[6] = ((i / 64) mod 2) = 1;
179   b[7] = ((i / 128) mod 2) = 1;
180
181   b[8] = ((i / 256) mod 2) = 1;
182   b[9] = ((i / 512) mod 2) = 1;
183   b[10] = ((i / 1024) mod 2) = 1;
184   b[11] = ((i / 2048) mod 2) = 1;
185   b[12] = ((i / 4096) mod 2) = 1;
186   b[13] = ((i / 8192) mod 2) = 1;
187   b[14] = ((i / 16384) mod 2) = 1;
188   b[15] = ((i / 32768) mod 2) = 1;
189
190   tel;
```

Jeux de Tests

Avec $p=3$ et $q=11$: (13,33), (17,33) et $m=31$:

```
[~/Desktop/repertoire/ArchMat/Real_mach_alg]
• base arhis " arhan-master - ./chiffrement 31 13 33
  Resultat chiffrement 25

[~/Desktop/repertoire/ArchMat/Real_mach_alg]
○ base arhis " arhan-master -
```

Luciole 1.75 - top16

Files Options Clocks Tools Inputs 0

<input type="checkbox"/> begin	m	exp	n	end_	q
<input type="checkbox"/> chM	?	?	?		?
<input type="checkbox"/> chEXP	etat_courant_0	etat_courant_1	etat_courant_2	etat_courant_3	etat_courant_4
<input type="checkbox"/> chN	etat_courant_5	etat_courant_6	etat_courant_7	etat_courant_8	etat_courant_9
input 0	etat_courant_10	Sig_chExp	Sig_chCode	Sig_chN	Sig_chM
	mise_1	selOp1_0	selOp1_1	selOp2	opUAL_0
	opUAL_1	Sig_C	Sig_Z	Sig_Odd	

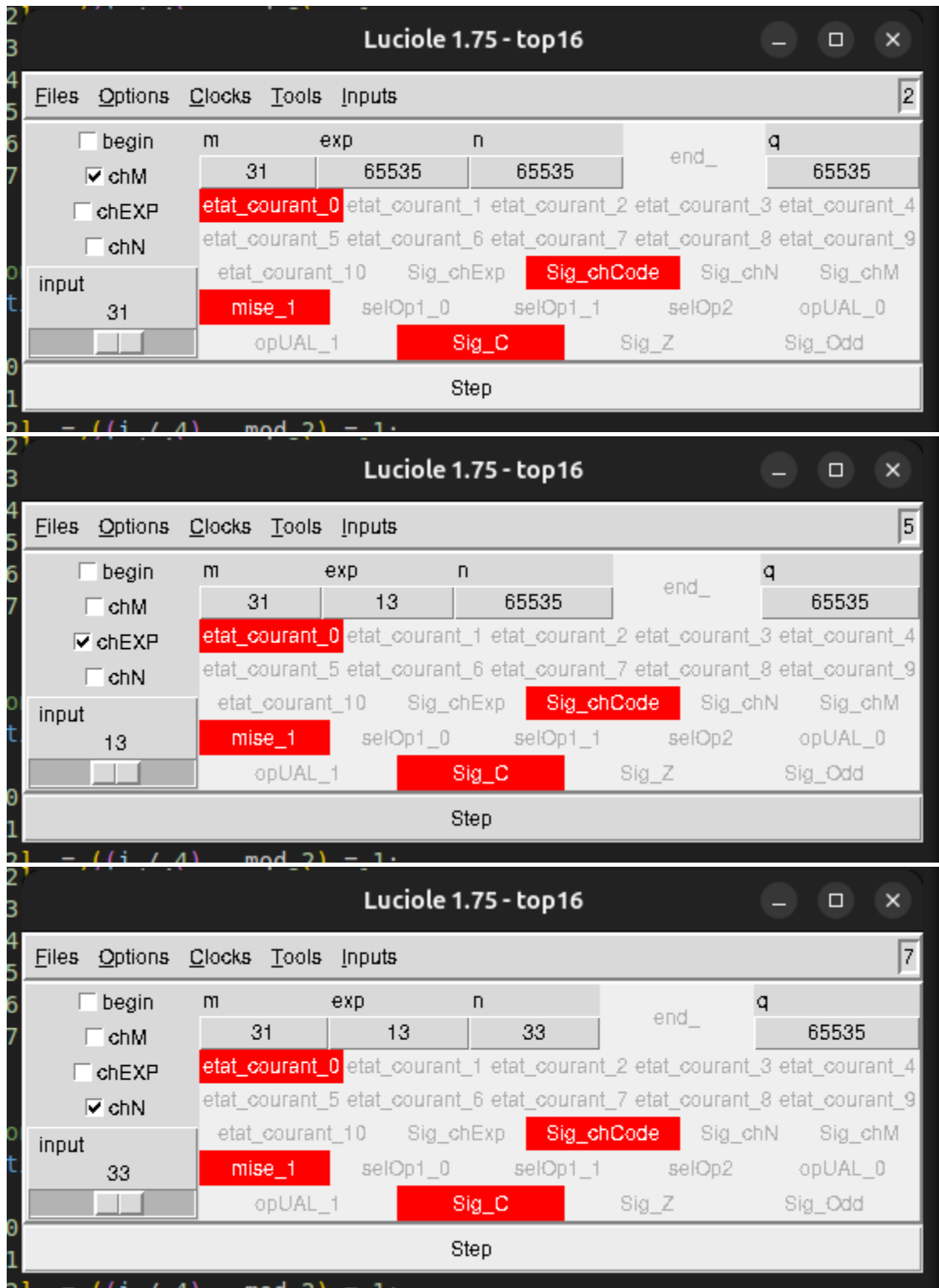
Step

Luciole 1.75 - top16

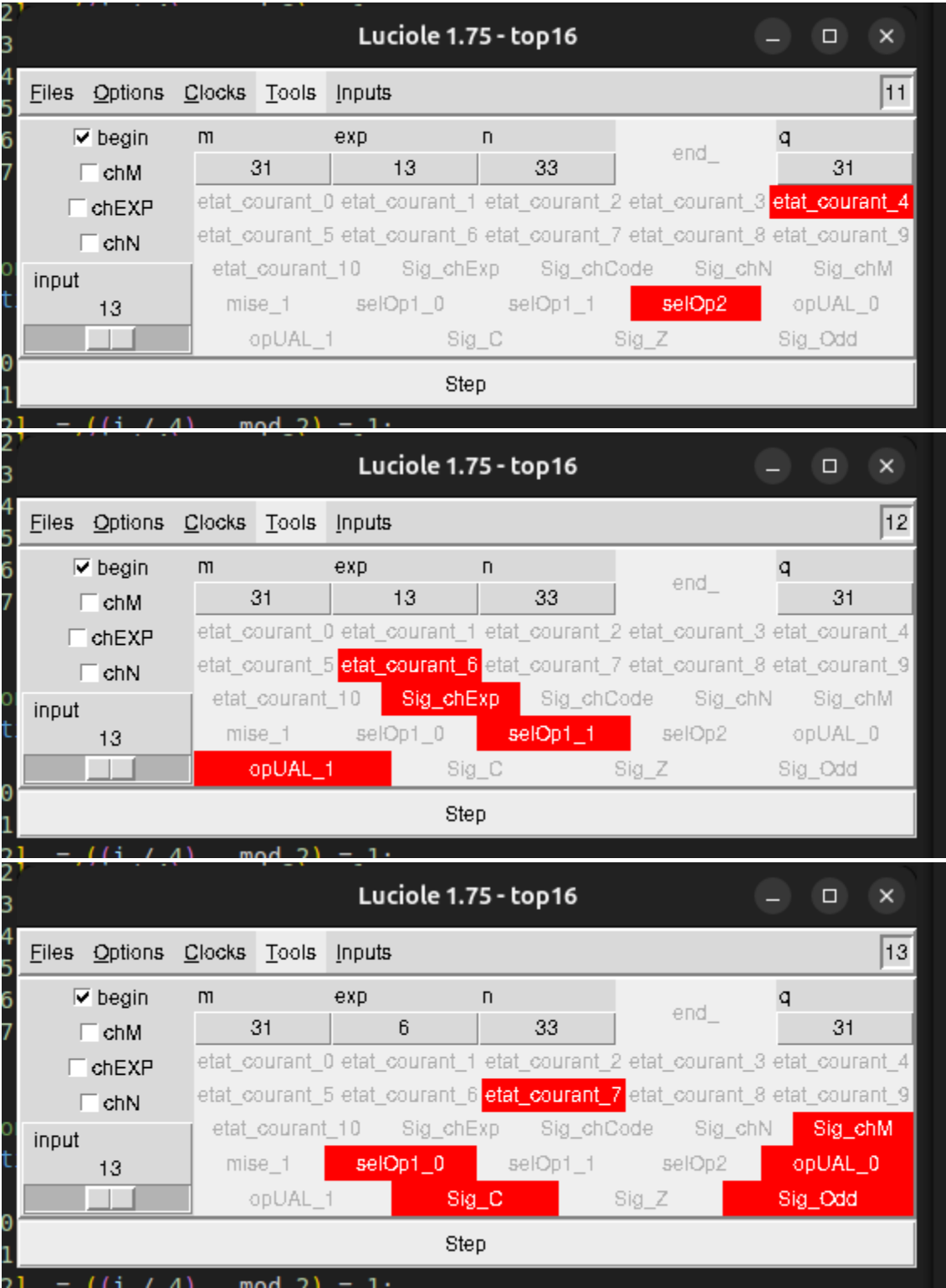
Files Options Clocks Tools Inputs 0

<input type="checkbox"/> begin	m	exp	n	end_	q
<input checked="" type="checkbox"/> chM	0	0	0		0
<input type="checkbox"/> chEXP	etat_courant_0	etat_courant_1	etat_courant_2	etat_courant_3	etat_courant_4
<input type="checkbox"/> chN	etat_courant_5	etat_courant_6	etat_courant_7	etat_courant_8	etat_courant_9
input 31	etat_courant_10	Sig_chExp	Sig_chCode	Sig_chN	Sig_chM
	mise_1	selOp1_0	selOp1_1	selOp2	opUAL_0
	opUAL_1	Sig_C	Sig_Z	Sig_Odd	

Step







Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

14

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
961	6	33		31
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

15

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
961	6	33		31
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

16

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
928	6	33		31
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

17

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m

exp

n

928

6

33

end_

q

etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4

etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9

etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM

mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0

opUAL_1 Sig_C Sig_Z Sig_Odd

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

18

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m

exp

n

895

6

33

end_

q

etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4

etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9

etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM

mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0

opUAL_1 Sig_C Sig_Z Sig_Odd

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

19

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m

exp

n

895

6

33

end_

q

etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4

etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9

etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM

mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0

opUAL_1 Sig_C Sig_Z Sig_Odd

Step

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

20

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m

exp

n

862

6

33

end_

q

etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4

etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9

etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM

mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0

opUAL_1 Sig_C Sig_Z Sig_Odd

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

21

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m

exp

n

862

6

33

end_

q

etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4

etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9

etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM

mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0

opUAL_1 Sig_C Sig_Z Sig_Odd

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

22

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m

exp

n

829

6

33

end_

q

etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4

etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9

etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM

mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0

opUAL_1 Sig_C Sig_Z Sig_Odd

Step



2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

26

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
763	6	33		31
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

27

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
763	6	33		31
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

28

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
730	6	33		31
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step





2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

Luciole 1.75 - top16

Files Options Clocks Tools Inputs 35

☒ begin
☐ chM
☐ chEXP
☐ chN

input 13

m	exp	n	end_	q
631	6	33		31
etat_courant_0	etat_courant_1	etat_courant_2	etat_courant_3	etat_courant_4
etat_courant_5	etat_courant_6	etat_courant_7	etat_courant_8	etat_courant_9
etat_courant_10	Sig_chExp	Sig_chCode	Sig_chN	Sig_chM
mise_1	selOp1_0	selOp1_1	selOp2	opUAL_0
opUAL_1	Sig_C	Sig_Z	Sig_Odd	

Step

Luciole 1.75 - top16

Files Options Clocks Tools Inputs 36

☒ begin
☐ chM
☐ chEXP
☐ chN

input 13

m	exp	n	end_	q
598	6	33		31
etat_courant_0	etat_courant_1	etat_courant_2	etat_courant_3	etat_courant_4
etat_courant_5	etat_courant_6	etat_courant_7	etat_courant_8	etat_courant_9
etat_courant_10	Sig_chExp	Sig_chCode	Sig_chN	Sig_chM
mise_1	selOp1_0	selOp1_1	selOp2	opUAL_0
opUAL_1	Sig_C	Sig_Z	Sig_Odd	

Step

Luciole 1.75 - top16

Files Options Clocks Tools Inputs 37

☒ begin
☐ chM
☐ chEXP
☐ chN

input 13

m	exp	n	end_	q
598	6	33		31
etat_courant_0	etat_courant_1	etat_courant_2	etat_courant_3	etat_courant_4
etat_courant_5	etat_courant_6	etat_courant_7	etat_courant_8	etat_courant_9
etat_courant_10	Sig_chExp	Sig_chCode	Sig_chN	Sig_chM
mise_1	selOp1_0	selOp1_1	selOp2	opUAL_0
opUAL_1	Sig_C	Sig_Z	Sig_Odd	

Step





2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

77

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
16	3	33		31
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

78

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

79

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step





2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

Luciole 1.75 - top16

Files Options Clocks Tools Inputs 86

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
16	3	33		430
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

Files Options Clocks Tools Inputs 87

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
16	3	33		397
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

Files Options Clocks Tools Inputs 89

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
16	3	33		364
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs90

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
16	3	33		364
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs91

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
16	3	33		331
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs109

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
16	3	33		34
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

21 = ((i / 4) mod 2) = 1:

21 = ((i / 4) mod 2) = 1:

21 = ((i / 4) mod 2) = 1:





Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

128

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

m	exp	n	end_	q
25	1	33		1
etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

129

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step

Luciole 1.75 - top16

FilesOptionsClocksToolsInputs

132

☒ begin

☐ chM

☐ chEXP

☐ chN

input

13

etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4				
etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9				
etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM				
mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0				
opUAL_1 Sig_C Sig_Z Sig_Odd				

Step





Pour les tests suivants, seuls les résultats et les états initiaux vous seront présentés.

```
[~/Desktop/repertoire/ArchMat/Real_mach_alg]
base ▶ arhis ▶ arhan-master - ./chiffrement 31 13 33
Resultat chiffrement 25

[~/Desktop/repertoire/ArchMat/Real_mach_alg]
base ▶ arhis ▶ arhan-master - ./chiffrement 25 17 33
Resultat chiffrement 31
```

Lucrative 1.7.5 - top16

Files Options Clocks Tools Inputs 8

☒ begin m exp n end_ q

☐ chM 25 17 33 1

☐ chEXP etat_courant_0 etat_courant_1 etat_courant_2 etat_courant_3 etat_courant_4

☐ chN etat_courant_5 etat_courant_6 etat_courant_7 etat_courant_8 etat_courant_9

input etat_courant_10 Sig_chExp Sig_chCode Sig_chN Sig_chM

17 mise_1 selOp1_0 selOp1_1 selOp2 opUAL_0

opUAL_1 Sig_C Sig_Z Sig_Odd

Step

```
[~/Desktop/repertoire/ArchMat/Real_mach_alg]
• base arhis arhan-master - ./chiffrement 35 13 77
Resultat chiffrement 63

[~/Desktop/repertoire/ArchMat/Real_mach_alg]
• base arhis arhan-master - ./chiffrement 63 37 77
Resultat chiffrement 35
```

The first screenshot shows the state of the machine at step 8. The interface includes a menu bar (Files, Options, Clocks, Tools, Inputs) and a status bar (Step 8). The main window displays a table of variables and their values:

begin	m	exp	n	end_	q
<input checked="" type="checkbox"/>	35	13	77		1
<input type="checkbox"/> chM	etat_courant_0	etat_courant_1	etat_courant_2	etat_courant_3	etat_courant_4
<input type="checkbox"/> chEXP	etat_courant_5	etat_courant_6	etat_courant_7	etat_courant_8	etat_courant_9
<input type="checkbox"/> chN	etat_courant_10	Sig_chExp	Sig_chCode	Sig_chN	Sig_chM
input	77	mise_1	selOp1_0	selOp1_1	selOp2
		opUAL_1	Sig_C	Sig_Z	opUAL_0
					Sig_Odd

The second screenshot shows the state of the machine at step 307. The interface is similar, but the status bar indicates Step 307. The table of variables is updated:

begin	m	exp	n	end_	q
<input checked="" type="checkbox"/>	42	0	77		63
<input type="checkbox"/> chM	etat_courant_0	etat_courant_1	etat_courant_2	etat_courant_3	etat_courant_4
<input type="checkbox"/> chEXP	etat_courant_5	etat_courant_6	etat_courant_7	etat_courant_8	etat_courant_9
<input type="checkbox"/> chN	etat_courant_10	Sig_chExp	Sig_chCode	Sig_chN	Sig_chM
input	77	mise_1	selOp1_0	selOp1_1	selOp2
		opUAL_1	Sig_C	Sig_Z	opUAL_0
					Sig_Odd

FilesOptionsClocksToolsInputs

8

☒ begin

☐ chM

☐ chEXP

☐ chN

input

77

m	exp	n	end_	q
63	37	77		1
etat_courant_0	etat_courant_1	etat_courant_2	etat_courant_3	etat_courant_4
etat_courant_5	etat_courant_6	etat_courant_7	etat_courant_8	etat_courant_9
etat_courant_10	Sig_chExp	Sig_chCode	Sig_chN	Sig_chM
mise_1	selOp1_0	selOp1_1	selOp2	opUAL_0
opUAL_1	Sig_C	Sig_Z		Sig_Odd

Step

FilesOptionsClocksToolsInputs

563

☒ begin

☐ chM

☐ chEXP

☐ chN

input

77

m	exp	n	end_	q
70	0	77		35
etat_courant_0	etat_courant_1	etat_courant_2	etat_courant_3	etat_courant_4
etat_courant_5	etat_courant_6	etat_courant_7	etat_courant_8	etat_courant_9
etat_courant_10	Sig_chExp	Sig_chCode	Sig_chN	Sig_chM
mise_1	selOp1_0	selOp1_1	selOp2	opUAL_0
opUAL_1	Sig_C	Sig_Z		Sig_Odd

Step