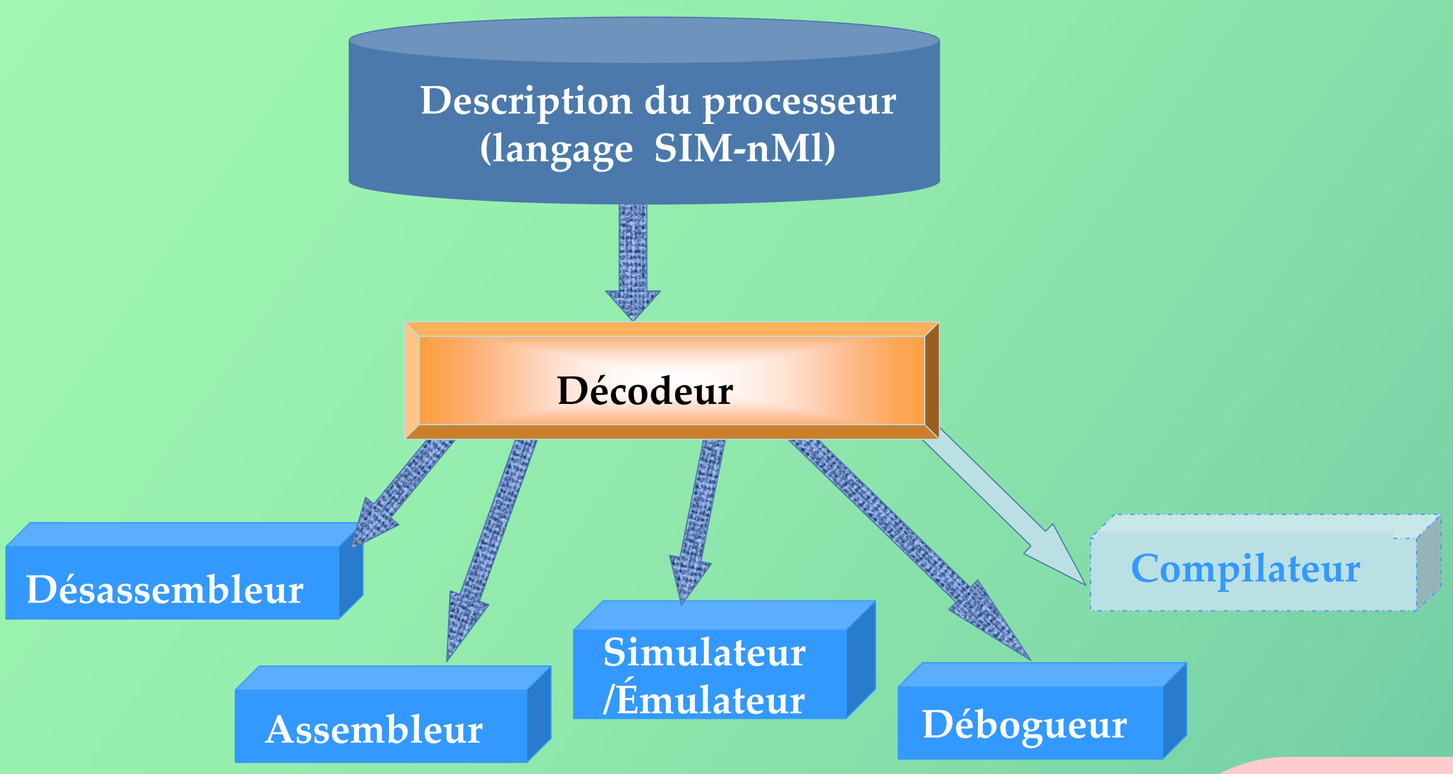
Tahiry Ratsiambahotra
ratsiam@irit.fr
IRIT, Université Paul Sabatier,
TOULOUSE

GÉNÉRATION DE SIMULATEURS FONCTIONNELS DE PROCESSEUR

Generator of Library of ISS (GLISS)

Structure du générateur



Modèle mémoire

PowerPC

```
let proc = "ppc"
let bit_order = "uppermost

//memory definition
let M_is_little = "MSR & (0x01<<MSR_LE)"
mem M       [ 32 , byte ]

//Registers definition
reg GPR     [ 2 ** REGS , slong ]
reg CR      [ 8 , half_byte ]

// Program counters
reg PIA     [ 1 , address ]
reg CIA     [ 1 , address ]
reg NIA     [ 1 , address ]
```

Modes d'adressage

HCS12X

```
//address mode definition
// for indexed adressing mode ,
 needed for encoding postbyte
  mode  IndX () = IX
                        syntax = "X"
                        image = "00"
  mode IndY () = IY
                        syntax = "Y"
                        image = "01"
  mode IndSP () = SP
                        syntax = "SP"
                        image = "10"
   mode  IndPC () = PC
                        syntax = "PC"
                        image = "11"

  mode RegIndIncr = IndX | IndY | IndSP
  mode RegIndSimple = IndX | IndY | IndSP | IndPC
  mode Register = RegIndSimple | RegOffsSimple
```

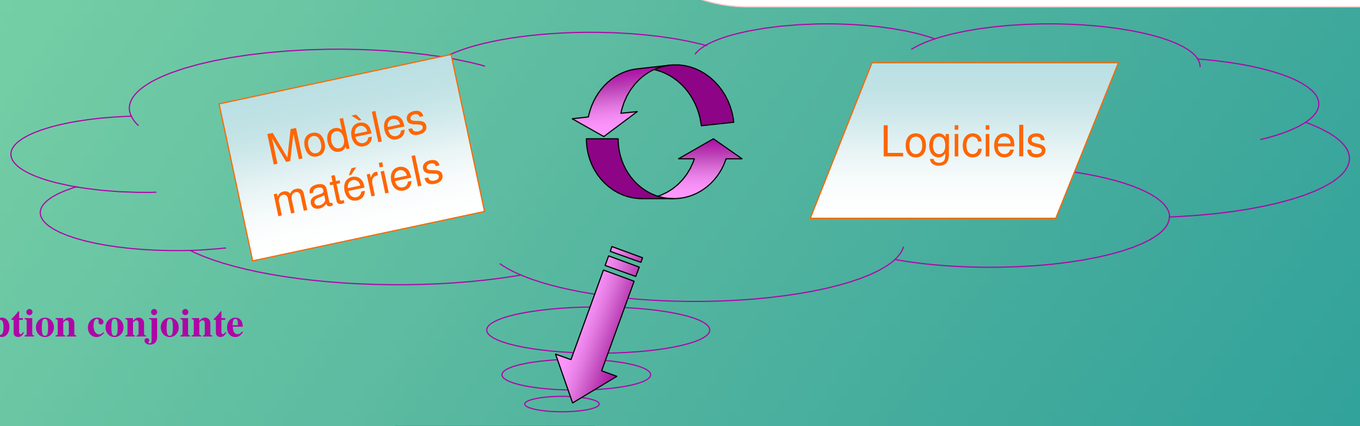
Description des instructions

ARM

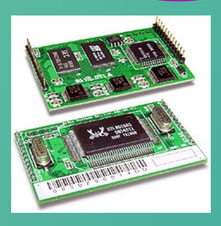
```
op instruction = ARM | THUMB
op ARM = dataProcessing | branch | LoadStore | LoadStoreM | interrupt
| semaphore |  multiply

// Data Processing Instructions
op dataProcessing = DataP
op DataP =  DataP_shr | DataP_imm

op DataP_imm(cond:card(4), opcode: card(4),sets:setS,rn:REG_INDEX, rd:
REG_INDEX, shifter_operand:rotatedImmediate)
predecode = {
        shifter_operand.predecode;
        switch (opcode)  {
              case 0: opcode = 'AND';
 ……….
             case 15: opcode = 'MVN'; };}
syntax = format("%s%s%s %s %s %s",opcode.syntax,cond.syntax,sets.syntax,
rd.syntax,rn.syntax,shifter_operand.syntax)
image = format("%4b001%4b%1b%s%s%s",cond,opcode,sets,rn.image,rd.image,
shifter_operand.image)
action = {if (CONDITION == 1) then
                        switch(opcode) {
                          case 0:   ANDM(rd,rn,shifter_operand,sets);
                        …….
                          case 15:  MVNM(rd,shifter_operand,sets); };
                        endif; }
```



- ✓**Conception conjointe**
- ✓**Test**
- ✓**Validation**



➢**Intégration**