UGH Inputs

Ivan Augé

Outline

1 User level

2 Internal level

Input example : Behavioral

```
#include <ugh.h>
/* communication channels */
ugh_inChannel32  instream;
ugh_outChannel32 outstream;
/* registers */
uint32 x,y;
/* behavior */
void ugh_main()
{
  while (1) {
    ugh_read(instream,&x);
    ugh_read(instream,&y);
    while (x!=y) {
      if (x<y) y = y - x ;
      else     x = x - y ;
    }
    ugh_read(outstream,&x);
  }
}
```

Main topic of behavioral input

Control

- Functions are inlined
- All control statement are supported : while, for, switch, if, break, continue, goto.
- WIRED : for wiring the if statement in DP.
  `if ( (cond) && WIRED) ...` $\Longrightarrow$ `if ( WIRED) ...`
  - Expanded to 1 when compiled for execution
  - Expanded to an external int variable when compiled for synthesis

  Can be replaced by a pragma

Main topic of behavioral input

Types & operators

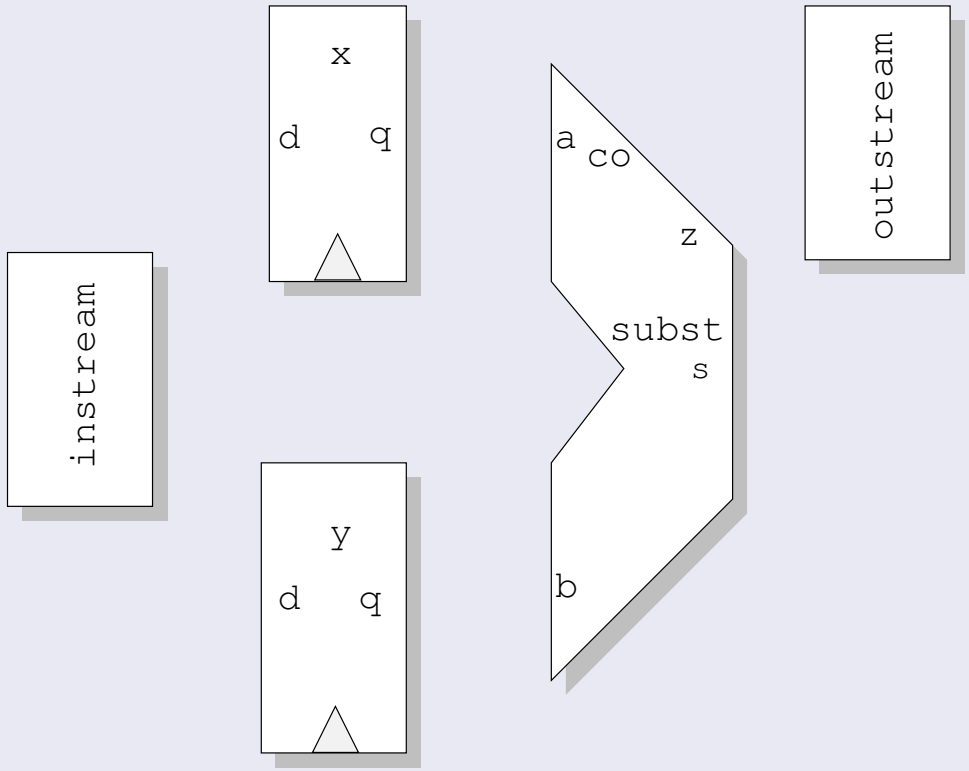
- wired types : intXX uintXX ugh_inChannelXX ugh_inChannelXX ($0 < XX < 129$, char=int8, short=int16, int=32, long=32 no float, no double)
- Structures, multi-dimensional arrays
- no pointer
- C operators : +, -, *, |, & ^, &&, ||, == ...
- C operators not supported : %, / execept for 2 power.
- added operator through wired function calls : ugh_read, ugh_write, ugh_smac, ugh_umac, ugh_shl/r, ugh_shra, ugh_rol/r, ugh_shlcat, ugh_catshr

Input example : Constraint

```
MODEL GCD(
      IN instream;
      OUT outstream)
{
DFF x, y;
SUB subst;




 }
```



Draft Data Path (DDP)

No bit size

Input example : Constraint

```
MODEL GCD(
      IN instream;
      OUT outstream)
{
DFF x, y;
SUB subst;




 }
```

```
ugh_inChannel32  instream;
ugh_outChannel32 outstream;
uint32 x,y;
void ugh_main() {
  while (1) {
    ugh_read(instream,&x);
    ugh_read(instream,&y);
    while (x!=y) {
      if (x<y) y = y - x ;
      else     x = x - y ;
    }
    ugh_read(outstream,&x);
} }
```

Link by name between register and variable

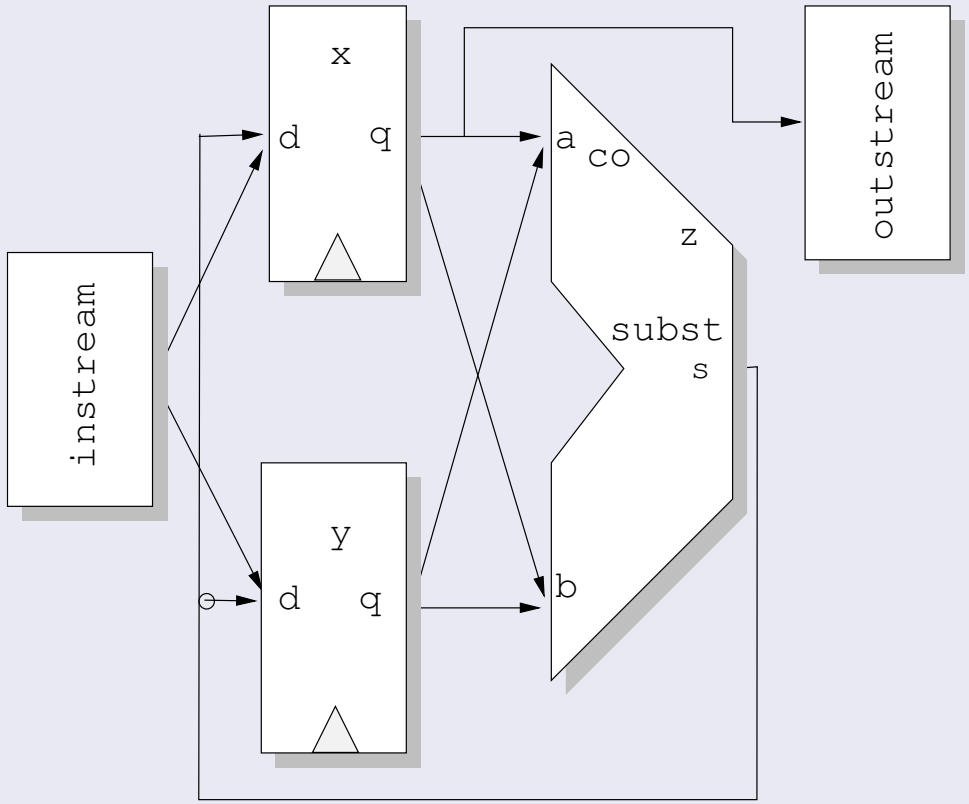
Allocation and binding of variables on memory elements done by the user

$\Longrightarrow$ Enough to give a dapapath close to user desiderata

Input example : Constraint

```
MODEL GCD(
      IN instream;
      OUT outstream)
{
DFF x, y;
SUB subst;

outstream = x.q;
subst.a = x.q, y.q;
subst.b = x.q, y.q;
x.d = subst.s, instream;
y.d = subst.s, instream;
 }
```



Adding arcs to target exactly a datapath

Outline

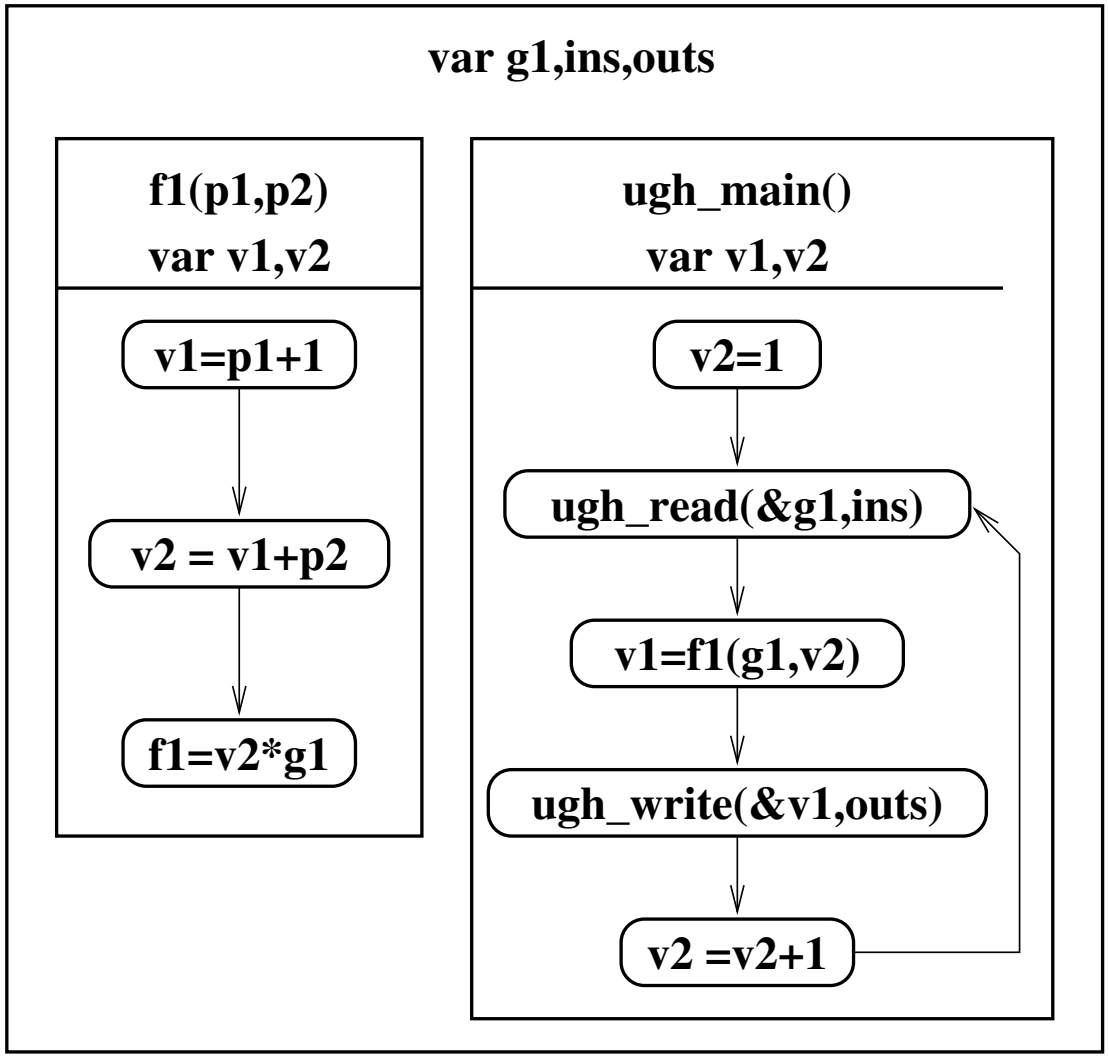
1 User level

2 Internal level

Behavioral

CDFG width extensions

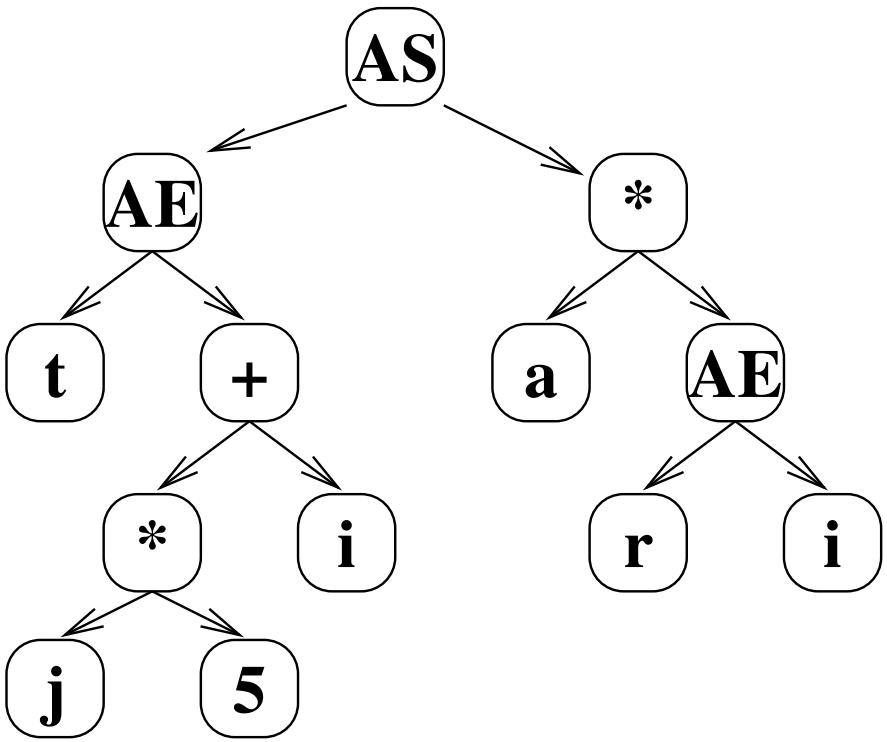
- Function definition
- Function call
- Variable signed or unsigned of N bits
- Array of 1 dimension
- Structure



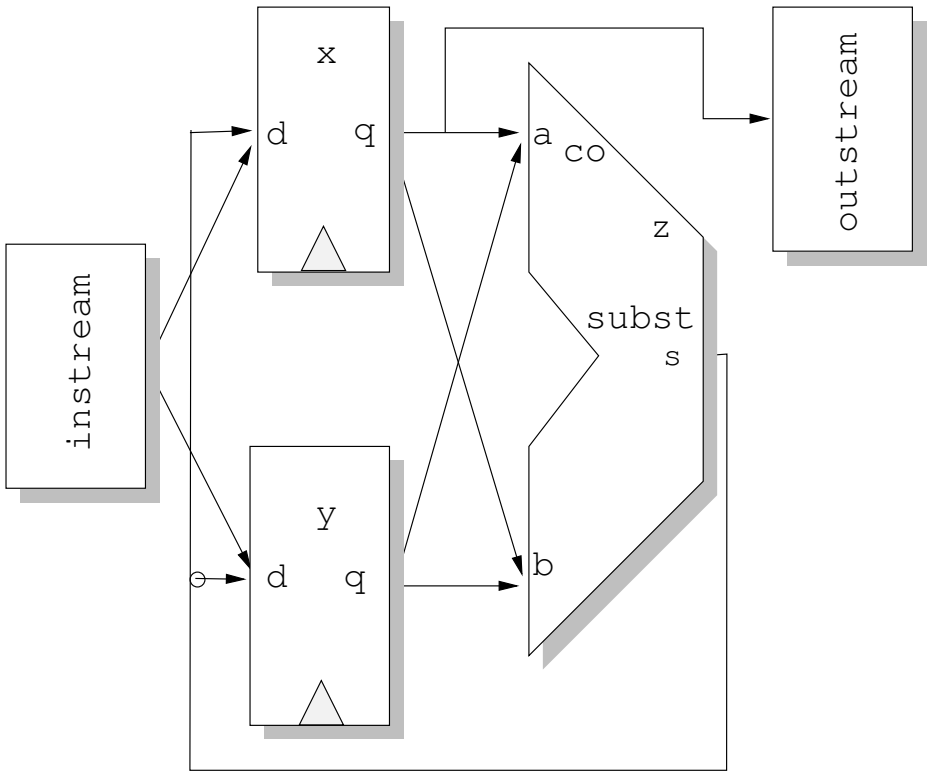
CDFG width extensions

- Function definition
- Function call
- Variable signed or unsigned of N bits
- Array of 1 dimension
- Structure

```
t[j][i] = a * r[i]
```



Constraint



- flat netlist
- node are predefined physical operators with ports
- vertices connect operator ports

$\Longrightarrow$ Doit-il entrer dans le xml ?