

# Interfaçage Générique d'IP

Tanguy Risset, Ludovic L'Hours, Romain Michard  
(Slide de Ludovic!)

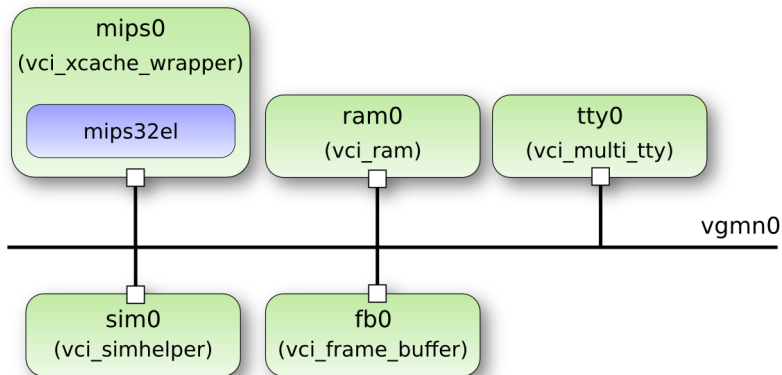
June 15, 2005

- Participation du CITI à SocLib
- Interfaçage d'IP
- Participation possible à Coach

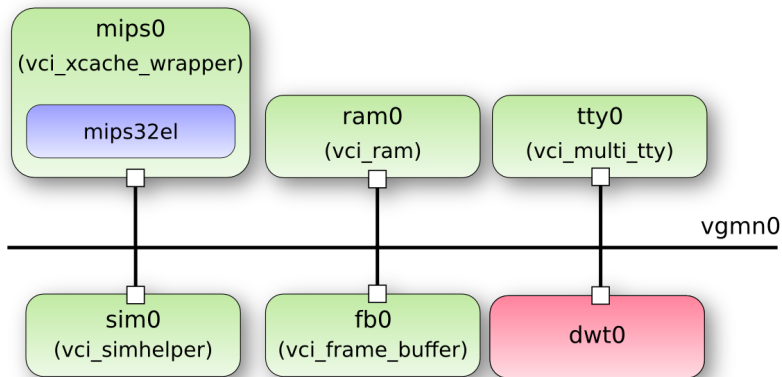
- Activité “systèmes embarqués” dans une équipe “réseau sans fils” (projet SWING)
- Effectif : environ 1 permanent
- Évolution des activités : de la synthèse de haut niveau (Alpha) vers des aspets plus système (NoC, interface, etc.)
- Participation aux projet SoCLib, OpenTLM

- Prototypage virtuel de plateformes hétérogènes multi-processeurs
- Différents modes de simulation : CABA ou TLM-(D)T
- Nombreux modèles de composants disponibles (ISS, mémoires, périphériques, IP, etc.)
- Plusieurs OS multitâches (pthread) et bientôt Linux
- Outillage divers : exploration (DSX), visualisation (SocView), etc.
- Projet ANR, 11 équipes académiques, 6 industriels

# Exemple de plateforme SoCLib



# Exemple de plateforme SoCLib

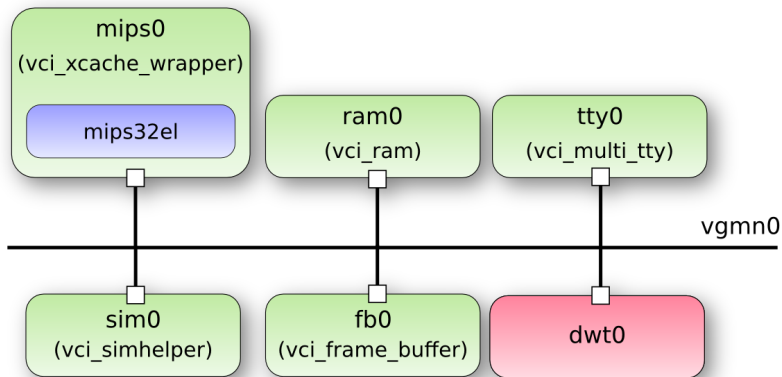


- Les différents processus de création d'IP génèrent des IP au comportement bien différents.
- Certaines IP exploitent des parallélisations d'algorithmes peu orthodoxes qui influent sur l'accès aux données.

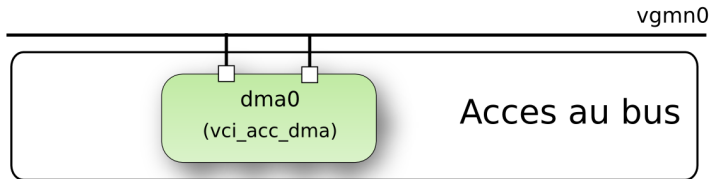
- Les différents processus de création d'IP génèrent des IP au comportement bien différents.
- Certaines IP exploitent des parallélisations d'algorithmes peu orthodoxes qui influent sur l'accès aux données.
- Comment alimenter en données ces IP?
- Comment les interfacer avec un système existant?



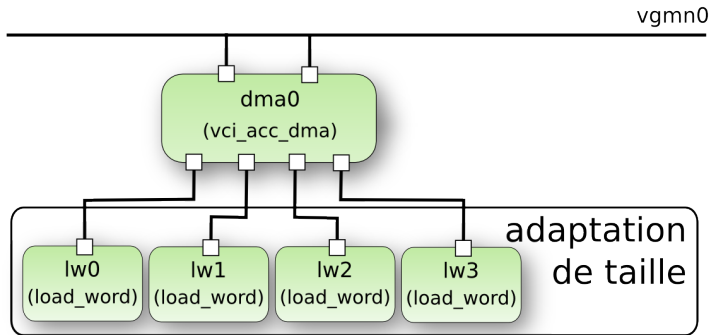
# Exemple de plateforme SoCLib



# Architecture de l'interface

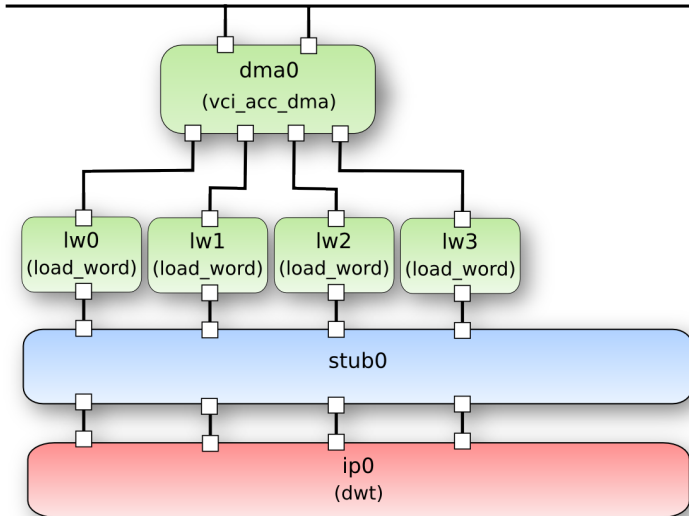


# Architecture de l'interface

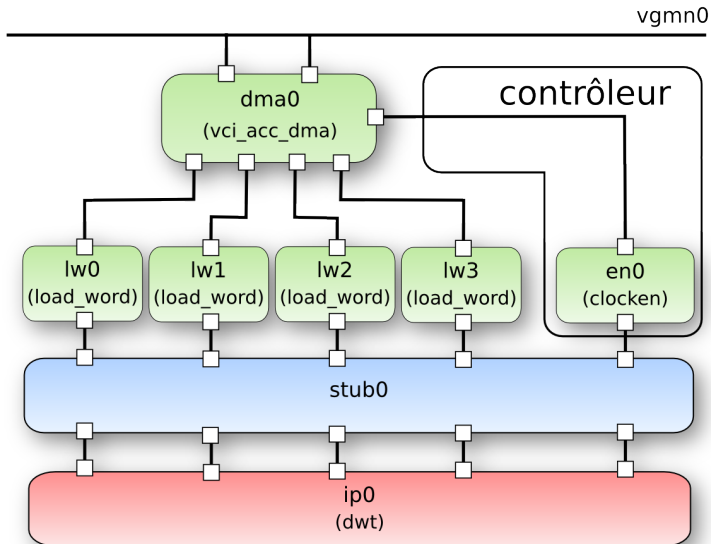


# Architecture de l'interface

vgmn0



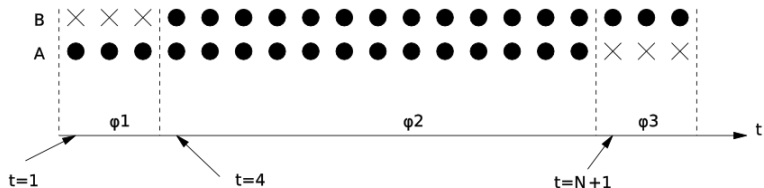
# Architecture de l'interface



# Architecture de l'interface(2)

- Moteur DMA programmable (Bus VCI)
- Adaptation en tailles de bus et endianness (optionel)
- Contrôleur de séquençement programmable (optionel)

# Programmation du Contrôleur



Modèle de programmation :

- pattern : activité sur les ports de l'IP
- phase : répétition d'un groupe de patterns
- super-phase : répétition d'un groupe de phases

Modèle de programmation :

- Unité de base : plage mémoire (adresse, longueur, FIFO, direction),
- Phase : groupe de plages mémoire dont l'accès peut être entrelacés
- Répétition de l'ensemble des phases avec un décalage (stride)
- Entrelacement des phases d'écriture et des phases de lecture

Modèle de programmation alternatif: MWMMR

- association FIFO logicielle/DMA
- Transfert de données non prévu statiquement mais dynamiquement
- Adapté au multi-processus
- Pourrait peut-être être combiné avec notre interface



- Contribution Soclib :
  - Modèle systemC caba de l'interface
  - Premiers résultats: un facteur 10 d'accélération (pour la partie accès des IP aux données)
- Contribution Coach :
  - Version VHDL (?)
  - Outil de génération de configuration de l'interface (driver soft et configuration hard).
- But de la participation:
  - Permettre de tester et évaluer rapidement les IP générées dans un environnement (par exemple SocLib)

## exemple de software driver: Clock enable

```
static void configure_clock_enable() {

    printf("Configuring clock-enable controller\n\n");
    unsigned int pattern = 0;
    unsigned int phase   = 0;
    soclib_io_set(base(IP), CTRL_RESET_REG, 0);

    // We got 3 phases and only 1 superphase
    // Phase 0
    set_ctrl(base(IP), PATTERN(pattern++),
             INPUT_OUTPUT_PATTERN(1, 0));
    set_ctrl(base(IP), PHASE(phase++), LENGTH_COUNT(1, 3));

    // Phase 1
    set_ctrl(base(IP), PATTERN(pattern++),
             INPUT_OUTPUT_PATTERN(1, 1));
    set_ctrl(base(IP), PHASE(phase++), LENGTH_COUNT(1, 5));

    // Phase 2
    set_ctrl(base(IP), PATTERN(pattern++),
```

## exemple de software driver: DMA

```
static void configure_and_start_dma(unsigned int size,
                                    unsigned int from,
                                    unsigned int to1,
                                    unsigned int to2) {

    printf("Configuring DMA controller\n\n");
    soclib_io_set(base(IP), DMA_RESET_REG, 0);

    // Read configuration
    soclib_io_set(base(IP), DMA_INFO_REG,
                  DMA_INFO(DMA_READ, DMA_MEM_CONTIG, 0));
    soclib_io_set(base(IP), DMA_MEM_REG, from);
    soclib_io_set(base(IP), DMA_PHASE_REG, 0);
    soclib_io_set(base(IP), DMA_LENGTH_REG, (size*4)>>2);

    soclib_io_set(base(IP), DMA_READ_LOOP_REG,
                  DMA_LOOP_COUNT_STRIDE(1, 0));

    // Write configuration
    soclib_io_set(base(IP), DMA_INFO_REG,
```

## exemple de software driver: main.c

```
int main(void) {
    unsigned int size = 8;
    printf("Start\n\n");
    configure_clock_enable();
    configure_and_start_dma(size,
                            (unsigned int)from_buffer,
                            (unsigned int)dst_buffer[0],
                            (unsigned int)dst_buffer[1]);

    wait_dma();
    printf("Output values:\n\n");
    for (i = 0; i<size; i++) {
        printf("out0: %d      out1: %d\n",
              dst_buffer[0][i],dst_buffer[1][i]);
    }
    printf("End\n\n");
    exit(0);
    return 0;
}
```