



Méthodes et outils pour l'implantation d'algorithmes temps réel sur architecture multi-DSP

Thierry Grandpierre

(t.grandpierre@esiee.fr)

Collaboration ESIEE/INRIA



Plan



- **Contexte & objectifs**
- **Modélisation**
 - Architecture
 - Algorithme
 - Implantation
- **Optimisation de l'implantation**
- **Génération automatique d'exécutif**
- **Conclusion & perspectives**



Contexte



- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- Optimisation
- Génér. code
- Conclusion

- **Applications** : automobile, robotique mobile, télécom, traitement d'images...
- **Algorithmes** : contrôle-commande, TSI
- **Réactif** : stimulus - opération - réaction
- **Temps Réel** :
 - Temps de réaction borné : latence
 - Période des stimuli bornée : cadence
- **Distribué** : puissance calcul, modularité,
- **Embarqué** : minimisation des ressources

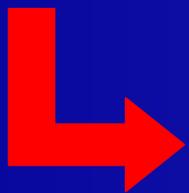


Objectifs



- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- Optimisation
- Génér. code
- Conclusion

- **Sécurité de conception** : sémantique langages synchrones + conservation des propriétés lors de l'implantation
- **Optimisation** : latence=cadence, taille archi.
- **Prototypage rapide** : réduction du cycle de développement
- **Génération de code de série**



Méthodologie



INRIA : Méthodologie Adéquation Algorithme Architecture



- Contexte
- AAA/SynDEX**
- Architecture
- Algorithme
- Implantation
- Optimisation
- Génération de code
- Conclusion

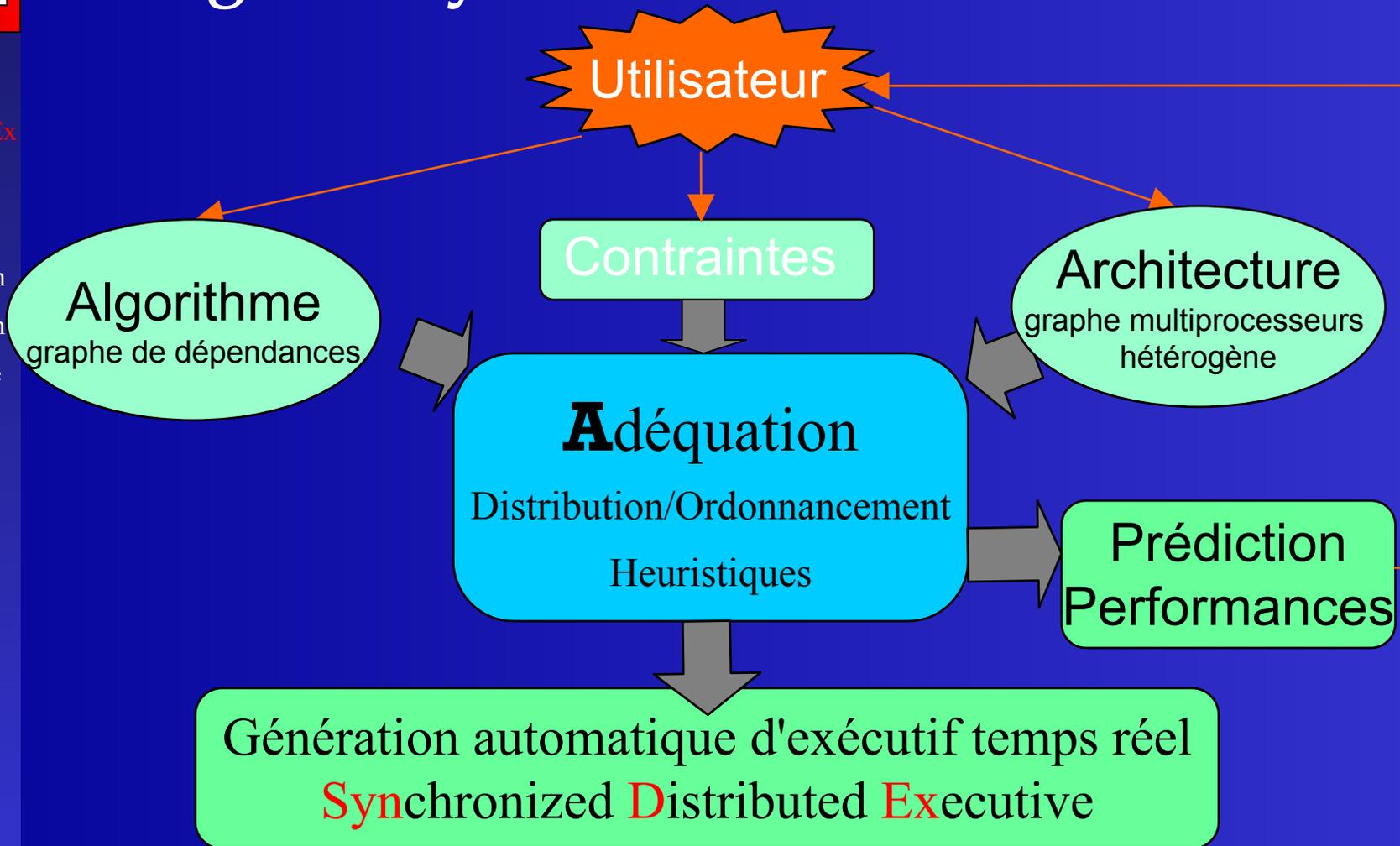
- **Approche globale**
- **Modèle unifié de graphes**
 - Algorithme : parallélisme potentiel
 - Architecture : parallélisme disponible
 - Implantation : transformations de graphes
- **Adéquation** : choix d'implantation optimisée
- **Génération automatique d'exécutifs** taillés sur mesure, basés sur ordonnancement hors-ligne des calculs et des communications



Méthodologie AAA et logiciel SynDEx



- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- Optimisation
- Génération de code
- Conclusion

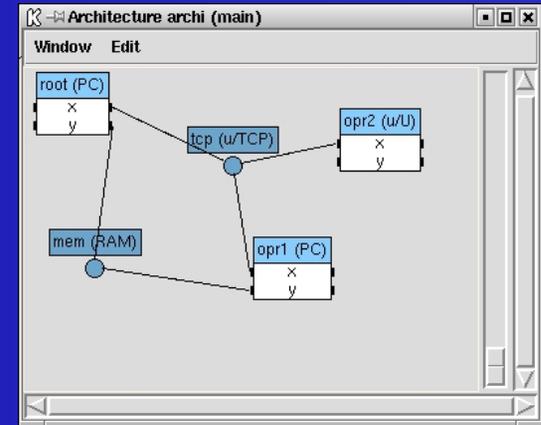
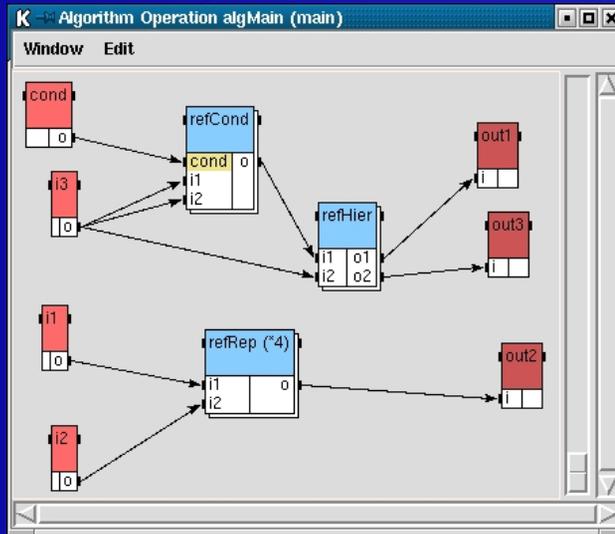




Méthodologie AAA : logiciel SynDEx



- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- Optimisation
- Génér. code
- Conclusion



```

K SynDEx-6.6.1 : v6news.sdx **
File Algorithm Architecture Constraints Adequation Code Help Debug

Reference : refCond
No software component

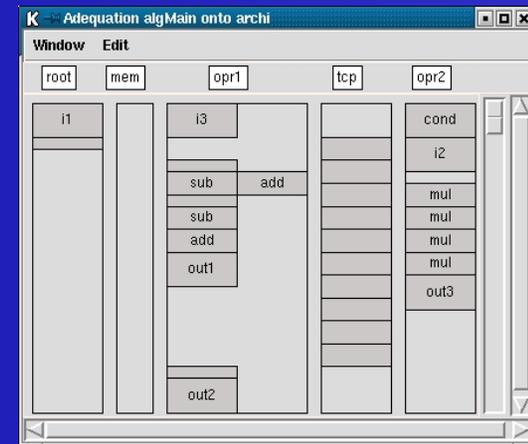
Definition : Operation defCond
Ports In
int cond[1]
int i1[1]
int i2[1]

Ports Out
int o[1]

Definition Operation algMain

Reference : refCond
No software component

Definition : Operation defCond
Ports In
int cond[1]
int i1[1]
    
```





Modèle d'architecture : graphe orienté



- Contexte
- AAA/SynDEx
- **Architecture**
- Algorithme
- Implantation
- Optimisation
- Génér. code
- Conclusion

• **Sommets (FSM) :**

- Opérateur : exécution séquentielle opns de calcul
- Communicateur : exéc. séquentielle opns comm.
- Mémoire avec ou sans arbitre, RAM ou SAM
- Bus/mux/démux avec ou sans arbitre :
 - Bus/mux/démux : sélect. une mémoire parmi plusieurs
 - Bus/mux/démux/arbitre : arbitre accès à mém. partagée

• **Arcs orientés :**

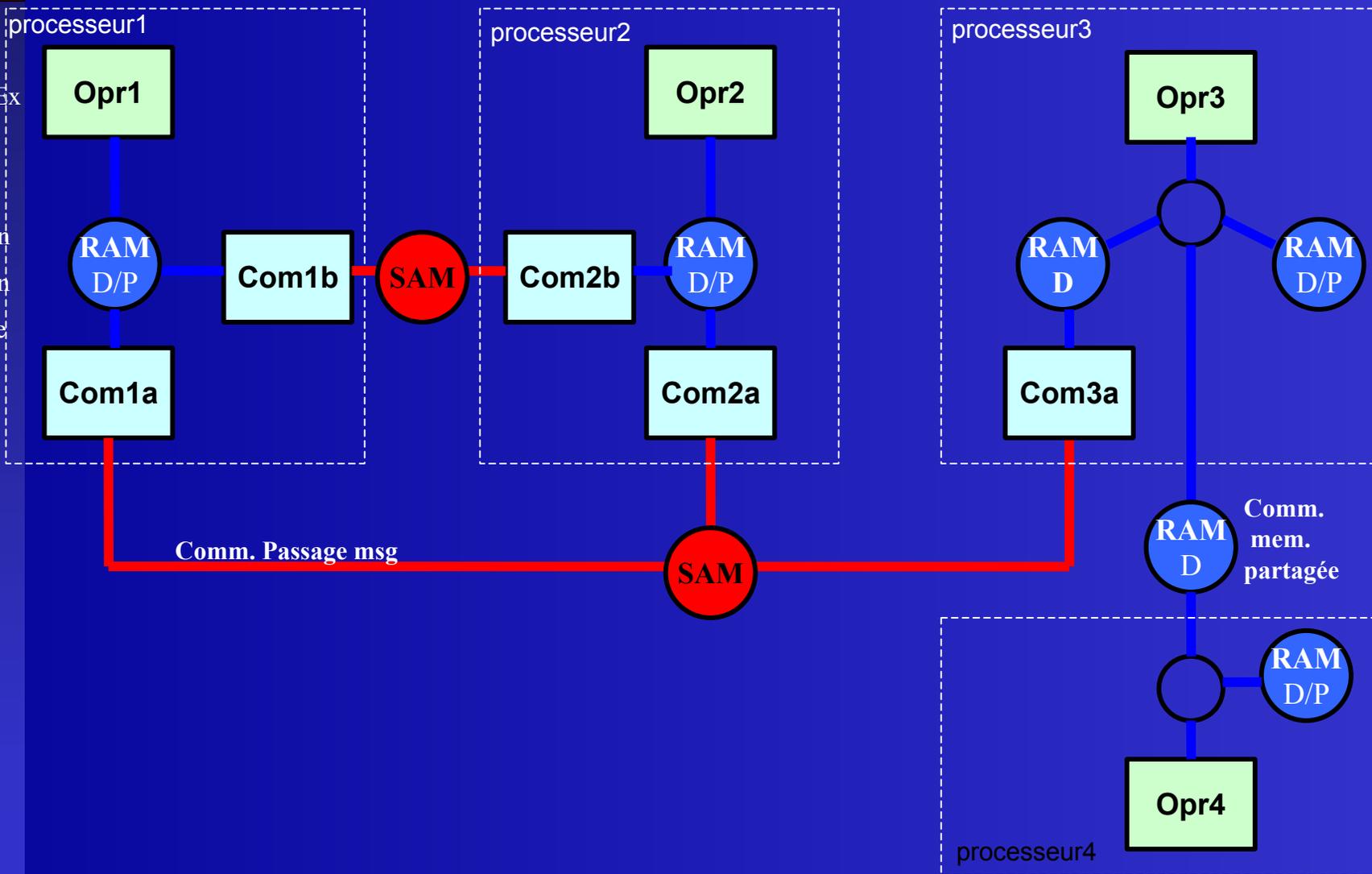
- transferts de données



Exemple de graphe d'architecture



- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- Optimisation
- Génér. code
- Conclusion

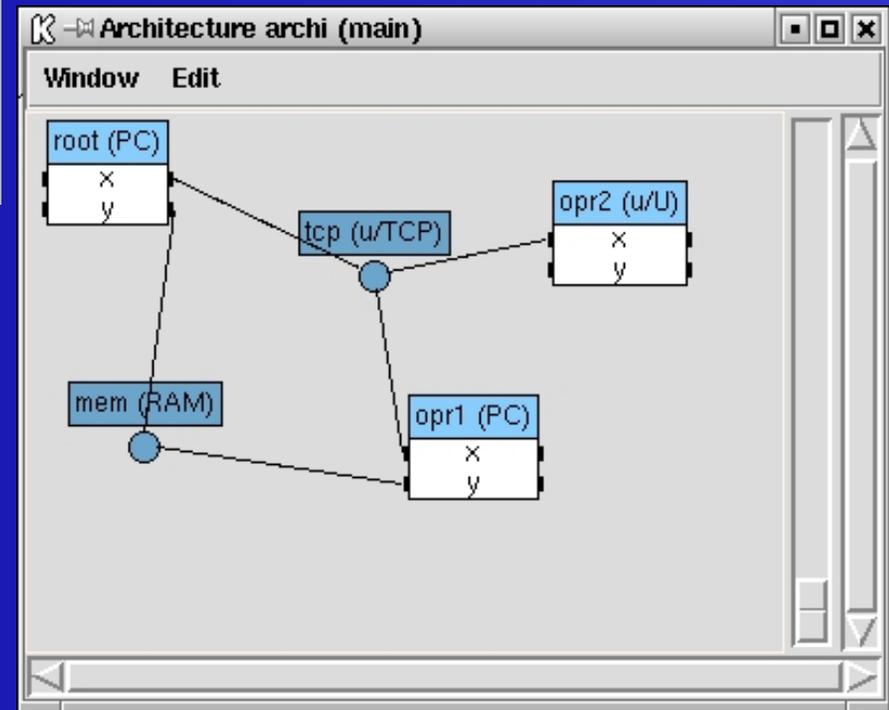
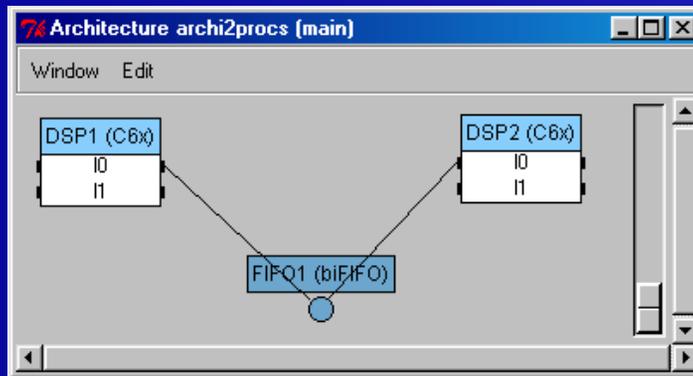




Architecture : capture d'écran SynDEx v6.6.2



- Contexte
- AAA/SynDEx
- **Architecture**
- Algorithme
- Implantation
- Optimisation
- Génér. code
- Conclusion



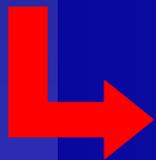


Modèle d'algorithme : hyper-graphe orienté



- Contexte
- AAA/SynDEx
- Architecture
- **Algorithme**
- Implantation
- Optimisation
- Génér. code
- Conclusion

- Sommet : opération conditionnée ou non
- Hyper-arc orienté : dépendance de données (ordre partiel d'exéc.) ou dép.de conditionnement
- **Répétition infinie d'un sous-graphe** (réactif)
- **Répétition finie d'un sous-graphe** (boucle)



Factorisation = Graphe flot de données conditionné

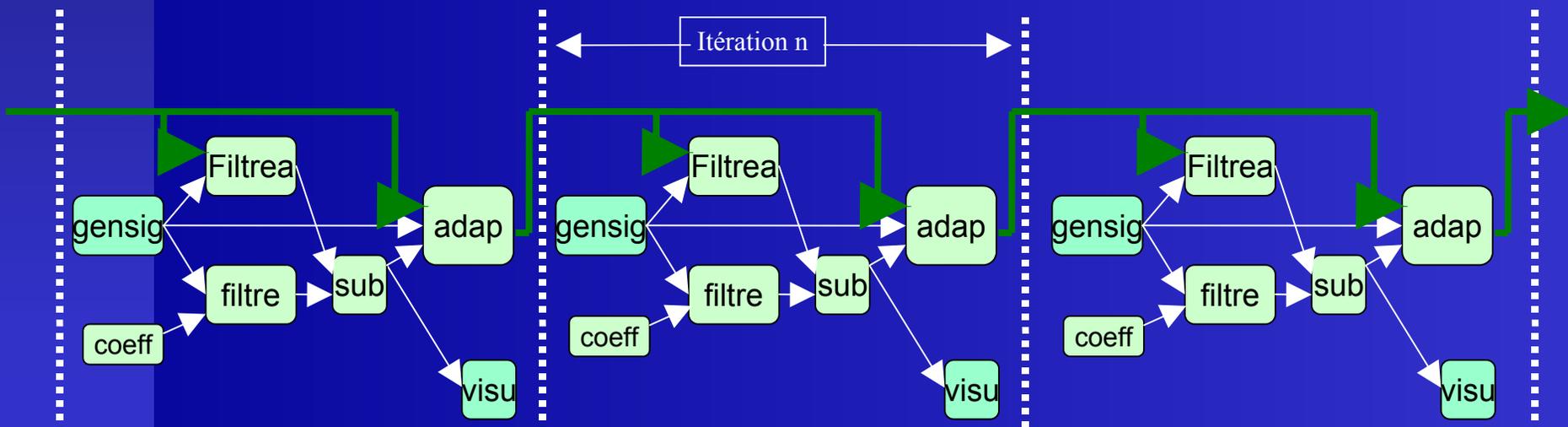
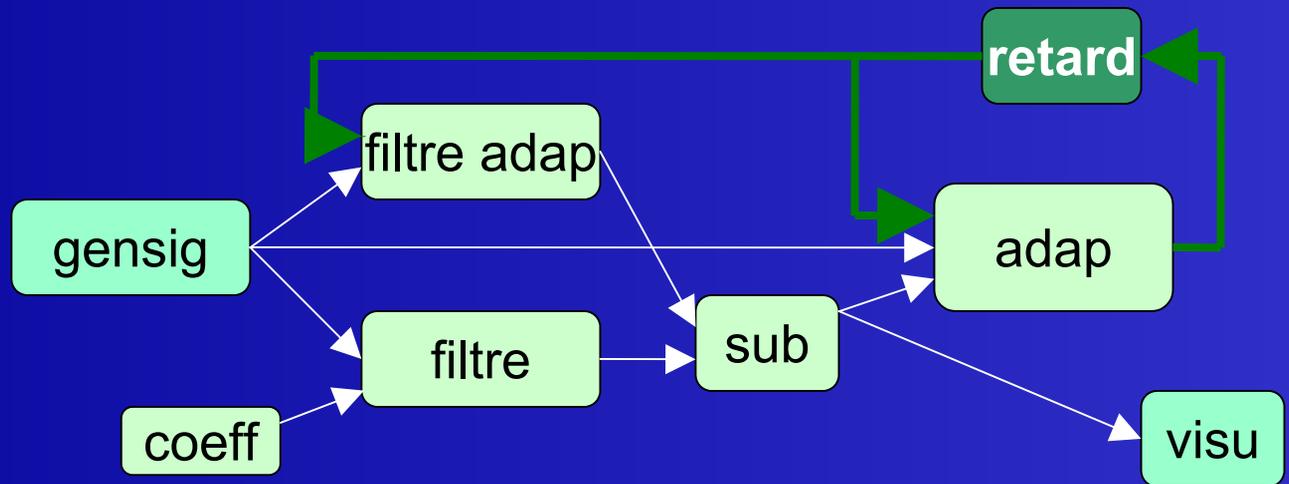
- **Issu des Langages Synchrones (via DC), de Scicos, AVS, CamlFlow**



Exemple d'algorithme : égaliseur adaptatif



- Contexte
- AAA/SynDEx
- Architecture
- **Algorithme**
- Implantation
- Optimisation
- Génér. code
- Conclusion

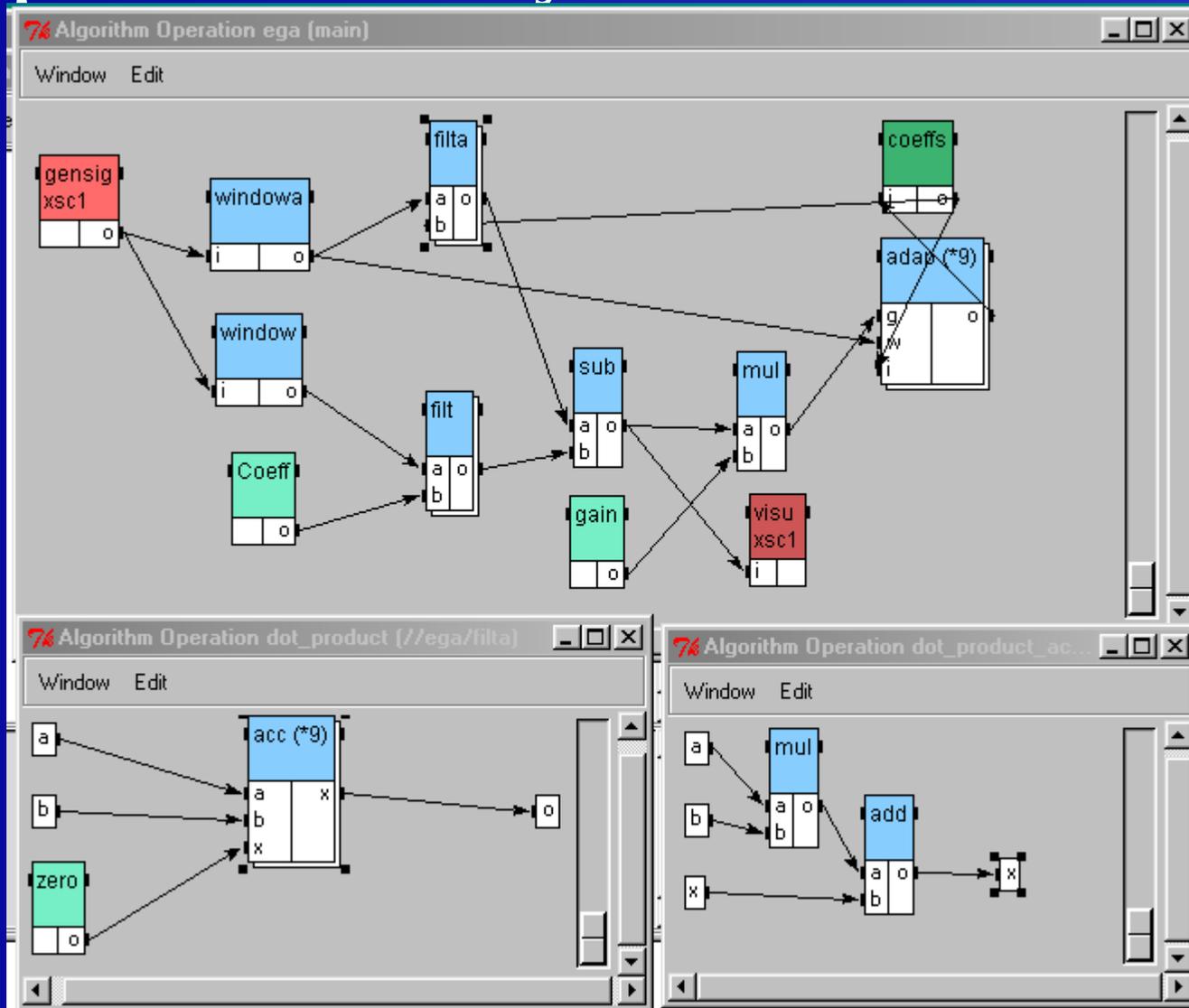




Algorithme : capture d'écran SynDEx v6.6.2



- Contexte
- AAA/SynDEx
- Architecture
- **Algorithme**
- Implantation
- Optimisation
- Génér. code
- Conclusion

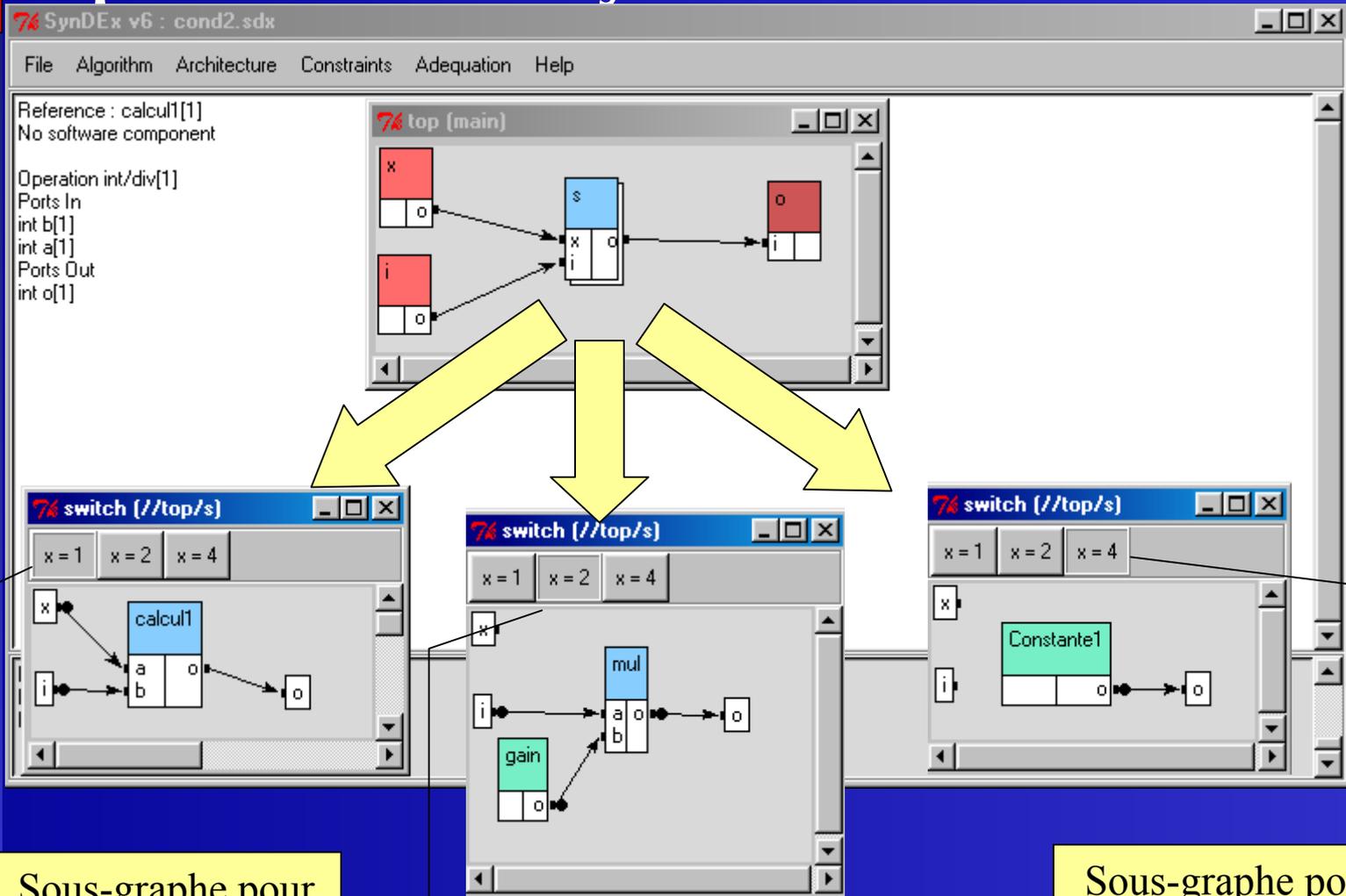




Algorithme avec conditionnement : capture d'écran SynDEx v6.6.2



- Contexte
- AAA/SynDEx
- Architecture
- **Algorithme**
- Implantation
- Optimisation
- Génér. code
- Conclusion



Sous-graphe pour Entrée x = 1

Sous-graphe pour Entrée x = 2

Sous-graphe pour Entrée x = 4



Modèle d'implantation



Toutes les implantations possibles décrites
par la composition de 3 relations binaires :

$(Gal, Gar) \xrightarrow{\text{rout o distrib o ordon}} (Gal', Gar)$

- **Routage** : calcul de toutes les routes
- **Distribution** : *allocation spatiale* :
 - ✓ Partition des opérations en fonction des opérateurs
 - ✓ Partition des arcs inter-partition en fonction des routes
 - ✓ Création et allocation : opns de comm., alloc./mém.
- **Ordonnancement** : *allocation temporelle*
 - ✓ Partition d'opérations : ordre partiel \rightarrow ordre total

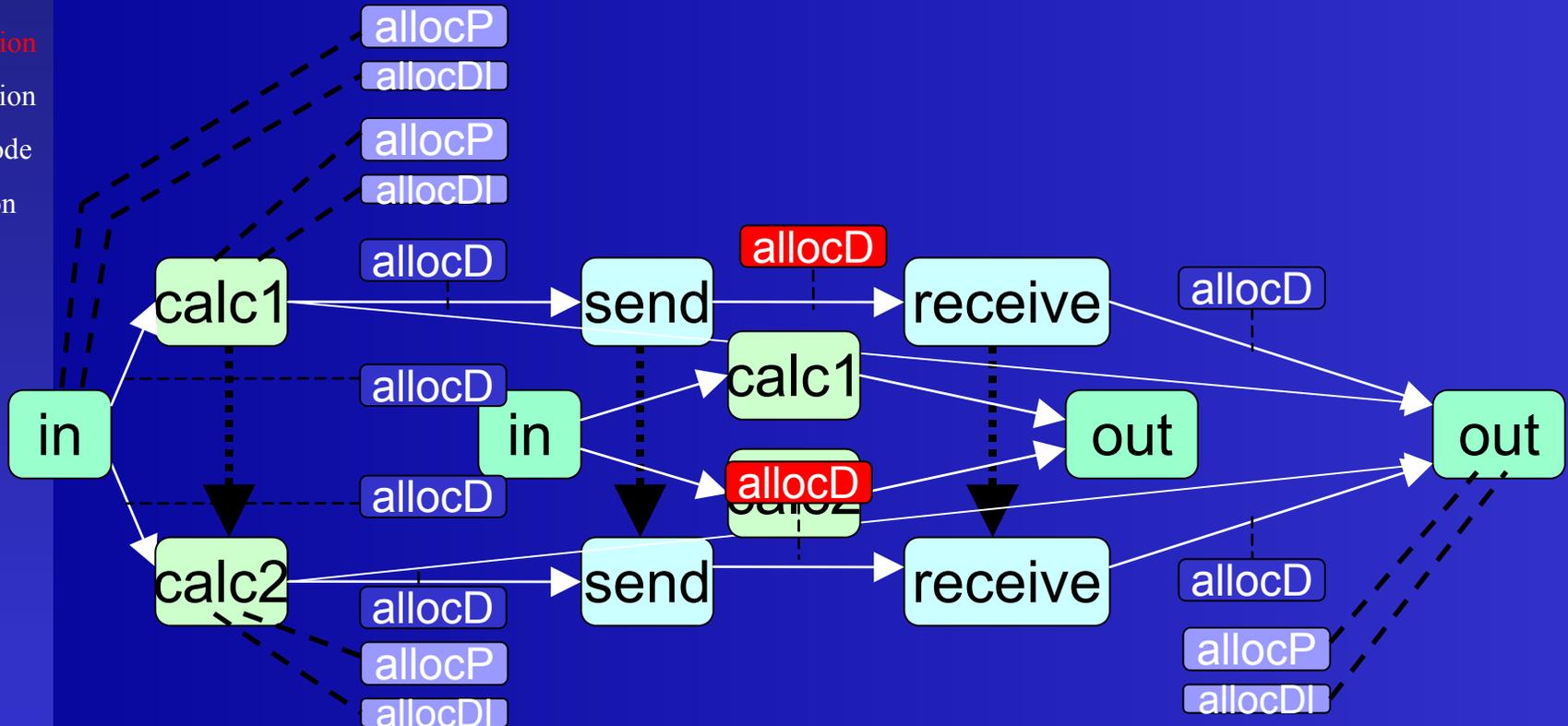
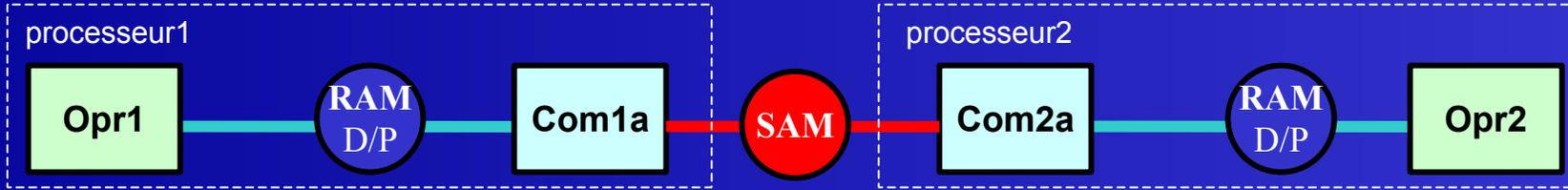
- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- **Implantation**
- Optimisation
- Génér. code
- Conclusion



Distribution / Ordonnancement



- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- **Implantation**
- Optimisation
- Génér. code
- Conclusion





Optimisation latence / ressources



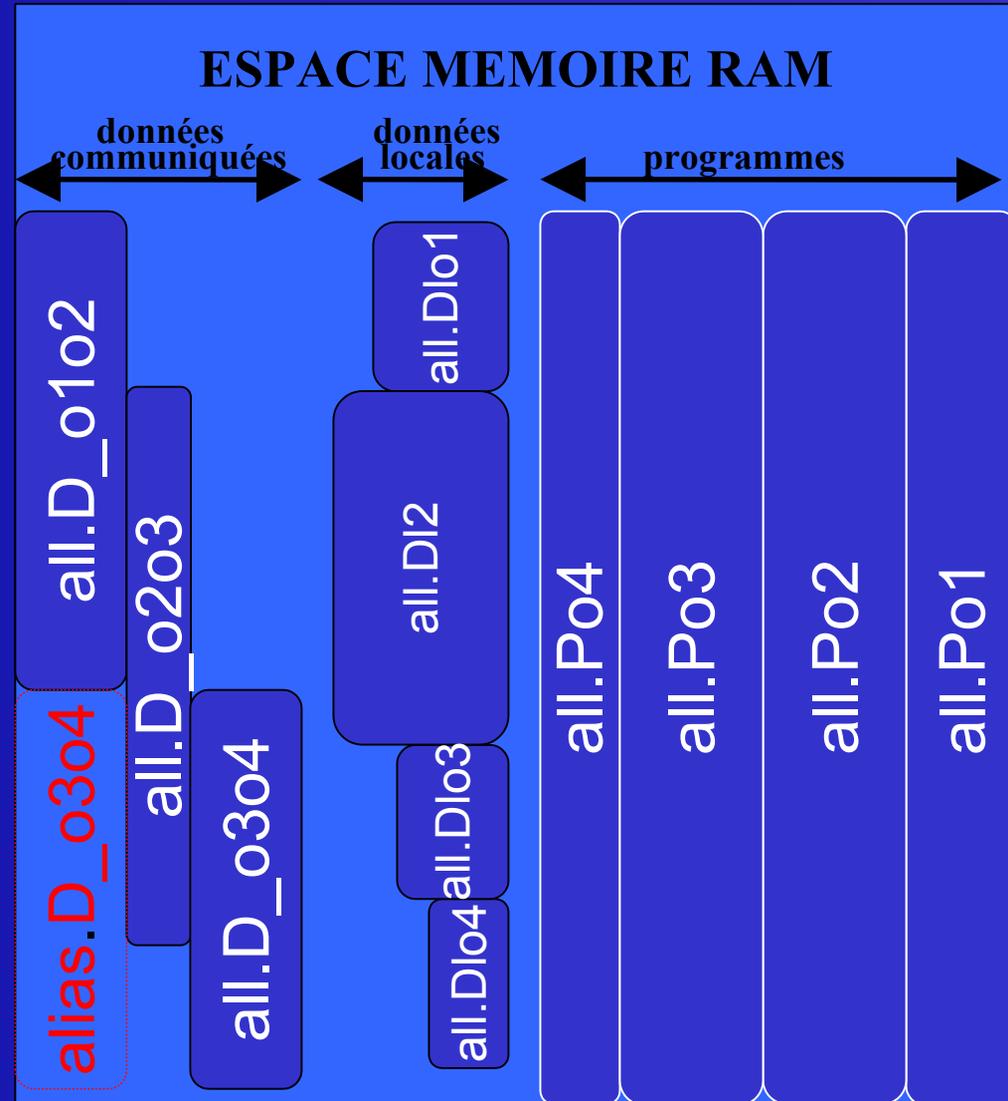
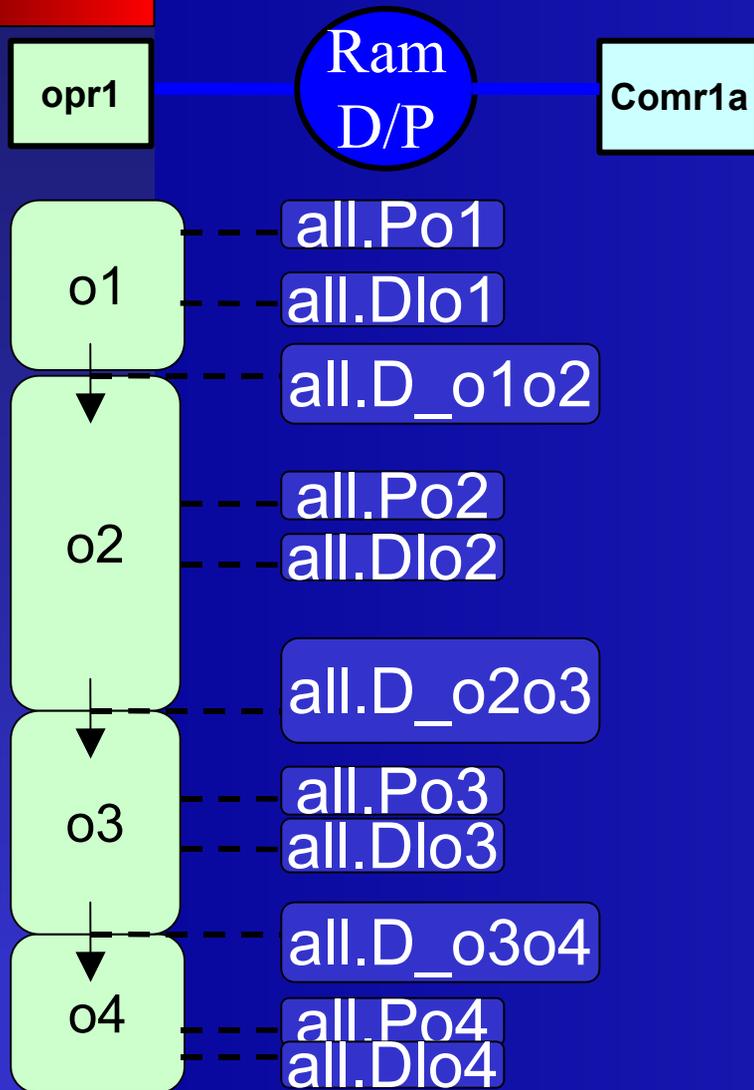
Adéquation : implantation optimisée = choix d'implantation **hors ligne** qui minimise latence

- C'est un problème **NP-complet** : heuristiques
 - Lentes itératives : voisinage local, global
 - Rapides (prototypage) : **gloutonne**
 - Fonction coût **pression d'ordonnancement**
 - Ordonnancement : **sans préemption**
- **Meta heuristique pour minimiser les ressources**

- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- **Optimisation**
- Génér. code
- Conclusion



Optimisation mémoire

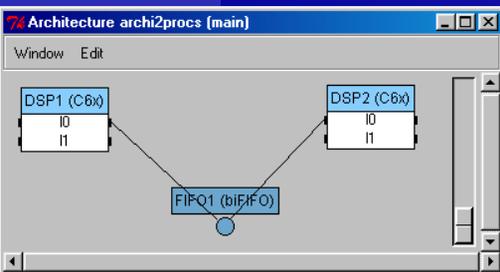
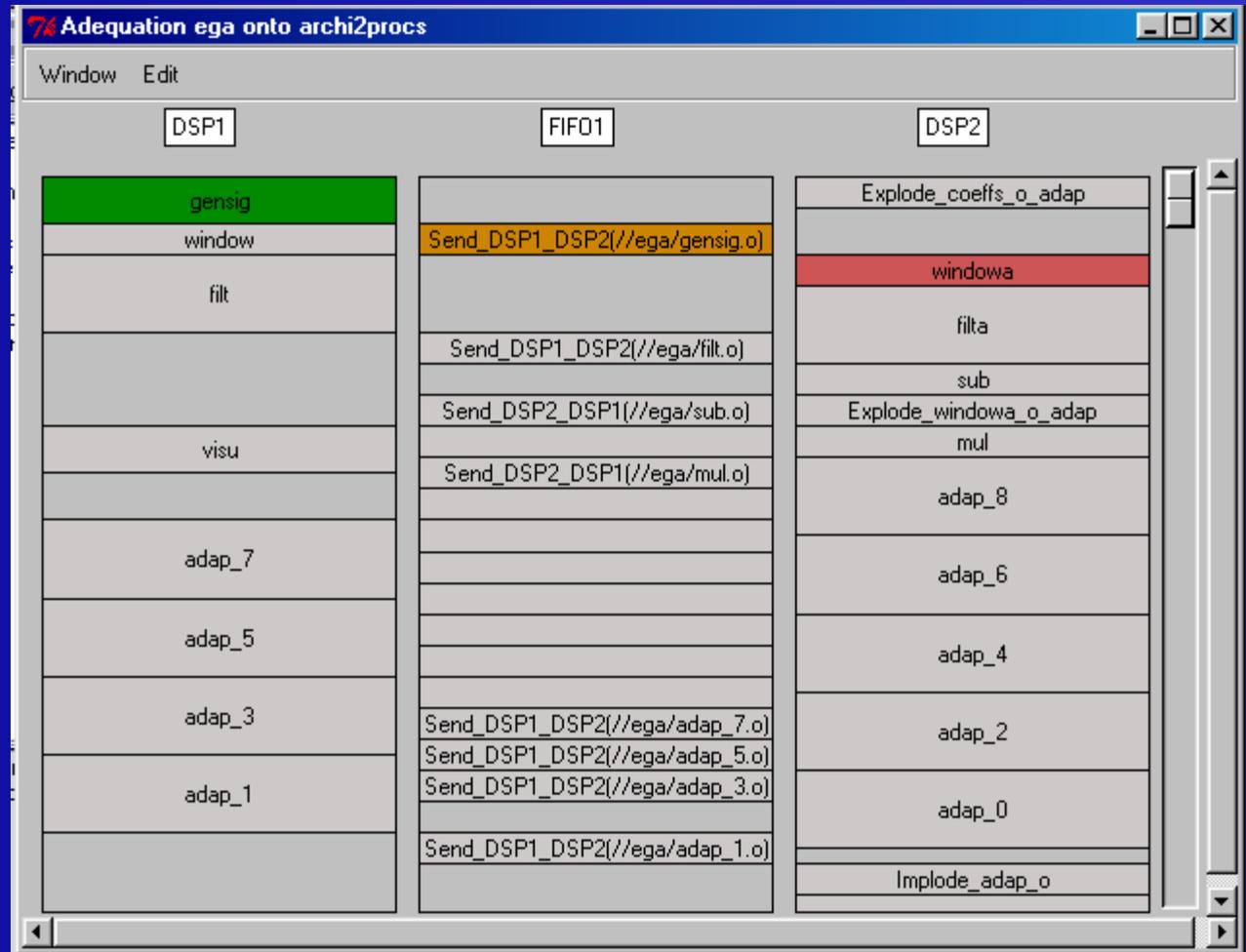




Implantation optimisée : capture d'écran SynDEx v6.6.2



- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- **Optimisation**
- Génér. code
- Conclusion





Génération d'exécutifs



Exécutif : supporte exéc. de l'algo. sur l'archi.= séquençement, gestion mém., comm., synchro.

- **Taillé sur mesure :**

- **Minimise surcoût** spatial et temporel
- Implant. conforme aux optimisations(com., mém.)
- **Sans inter-blocage**, garantie ordre partiel graphe algo.

- **Basé sur macro-code**

- **Indépendant** du processeur, portabilité

- **Basé sur des bibliothèques génériques dépendantes des processeurs (pas de RTOS)**

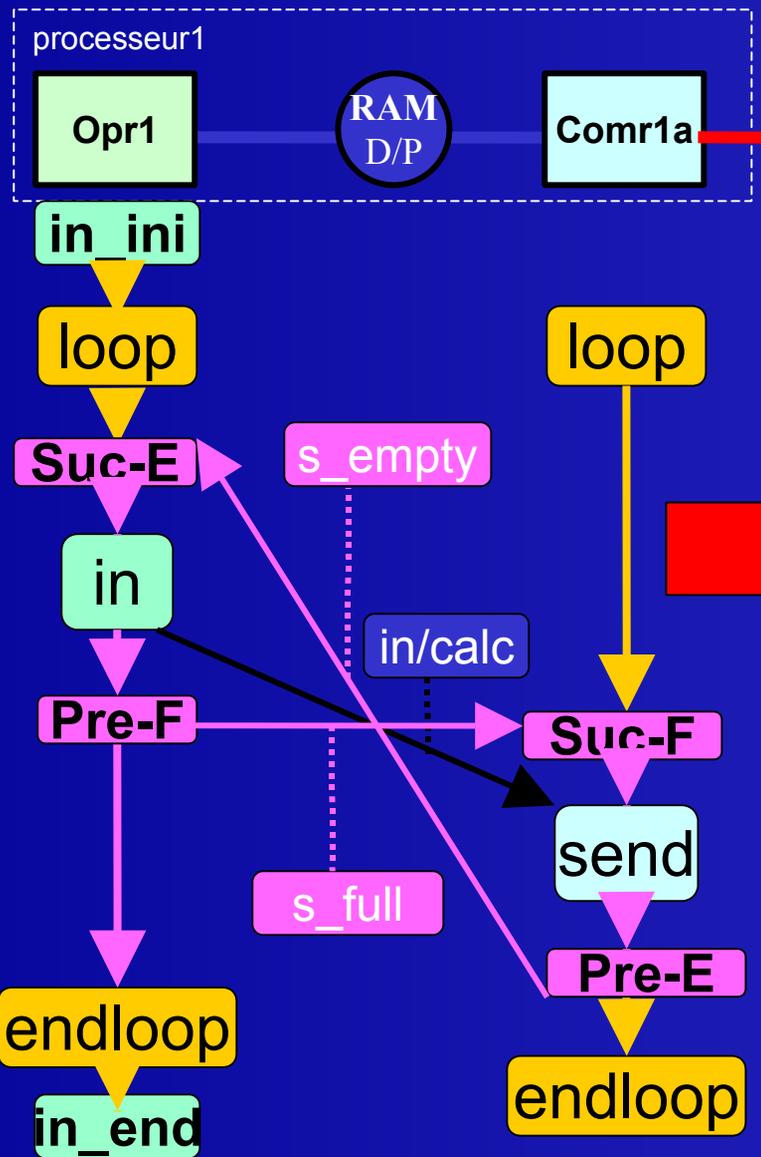
- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- Optimisation
- **Génér. code**
- Conclusion



Génération de macro code



- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- Optimisation
- Génér. code
- Conclusion



```

processor_( opr1, type opr1)
semaphores_(s_empty, s_full)
alloc_(type in/calc, in/calc..)
thread_(comr1a)
pre0(s_empty)
loop_
sucF(s_full)
send(in/calc)
preE(s_empty)
endloop_
endthread_
main_
in_ini()
spawn_thread(comr1a)
loop_
sucE(s_empty)
in(in/calc)
preF(s_full)
endloop_
in_end()
endmain_
endprocessor_

```



Compilation du macro code



```
processor_(opr1,.)  
semaphores(s_empty...)  
alloc_(type_in/calc...)  
thread_(com1ra)  
pre0(s_empty)  
loop  
sucF(s_full)  
send(in/calc)  
preE(s_empty)  
endloop  
endthread_  
main  
in_ini()  
  
spawn_thread(com1ra)  
loop  
sucE(s_empty)  
in(in/calc)  
preF(s_full)  
endloop  
in_end()  
endmain_  
endprocessor_
```

**BIBLIOTHEQUE
GENERIQUE
D'EXECUTIF
indépendante de
l'architecture
(syndex.m4x)**

**BIBLIOTHEQUE
GENERIQUE
D'EXECUTIF
spécifique à
l'architecture
(Unix.m4x, Tcp.m4x
Can.m4x, C6x.m4x)**

**MACRO
PROCESSEUR**

src

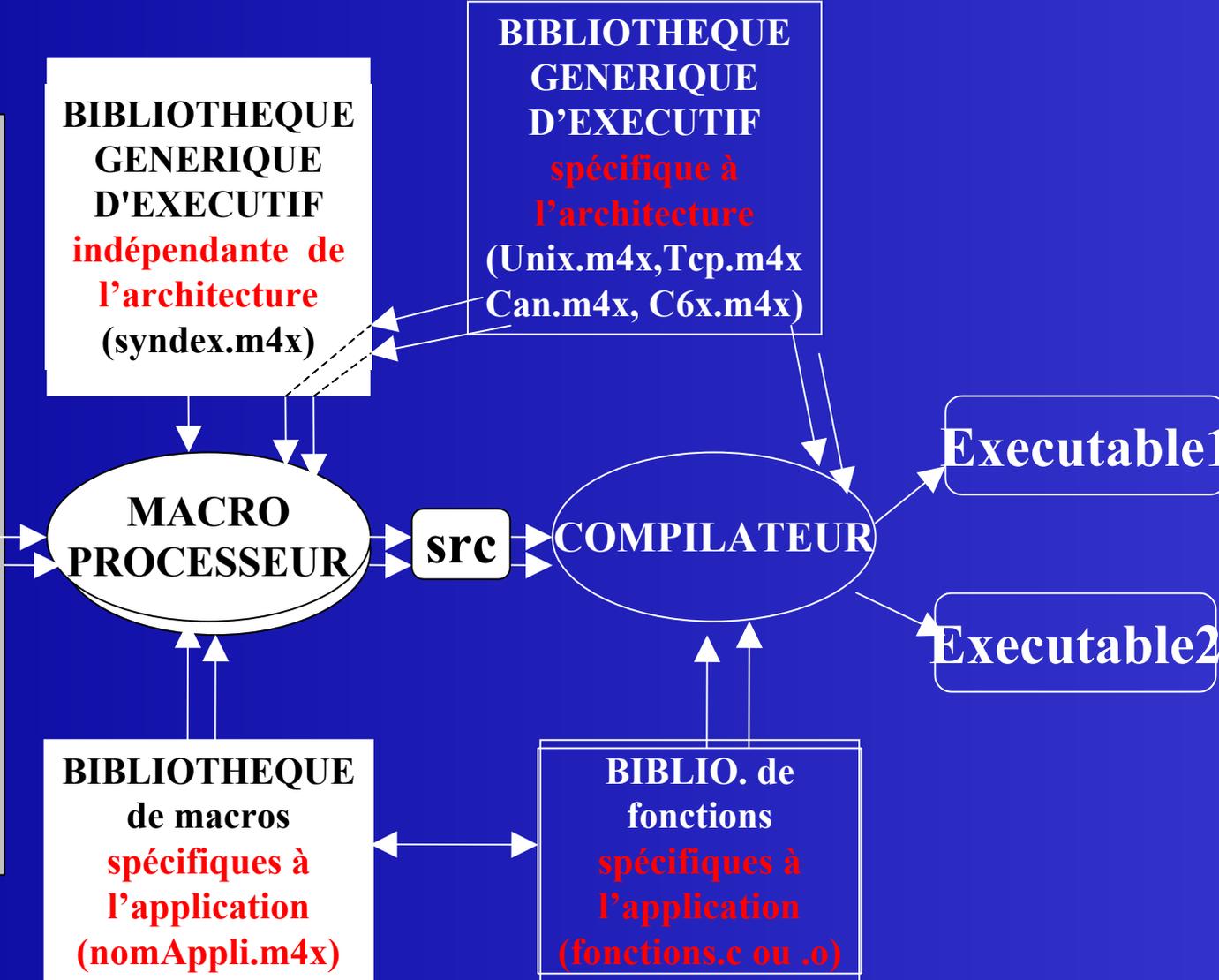
COMPILATEUR

Executable1

Executable2

**BIBLIOTHEQUE
de macros
spécifiques à
l'application
(nomAppli.m4x)**

**BIBLIO. de
fonctions
spécifiques à
l'application
(fonctions.c ou .o)**





Génération d'exécutif : capture d'écran SynDEx v6.6.2



- Contexte
- AAA/SynDEx
- Architecture
- Algorithmes
- Implantation
- Optimisation
- Génér. code
- Conclusion

```
74 SynDEx-6.6.2 : C:/Program Files/syndex-6.6.2/examples/ega/ega2/DSP
include(syndex.m4x)
include(float.m4x)
include(u.m4x)
processor_(C6x,DSP2,demoateme,
SynDEx-6.6.2 (c)INRIA 2002, 4/6/2003 08:45:50
)

semaphores_(
Semaphore_Thread_I0,
ega_adap_1_o_DSP2_I0_empty,
ega_adap_1_o_DSP2_I0_full,
ega_adap_3_o_DSP2_I0_empty,
ega_adap_3_o_DSP2_I0_full,
ega_adap_5_o_DSP2_I0_empty,
ega_adap_5_o_DSP2_I0_full,
ega_adap_7_o_DSP2_I0_empty,
ega_adap_7_o_DSP2_I0_full,
ega_Explode_windowa_o_adap_o_1_DSP2_I0_empty,
ega_Explode_windowa_o_adap_o_1_DSP2_I0_full,
ega_Explode_coefs_o_adap_o_1_DSP2_I0_empty,
ega_Explode_coefs_o_adap_o_1_DSP2_I0_full,
ega_Explode_windowa_o_adap_o_3_DSP2_I0_empty,
ega_Explode_windowa_o_adap_o_3_DSP2_I0_full,
ega_Explode_coefs_o_adap_o_3_DSP2_I0_empty,
ega_Explode_coefs_o_adap_o_3_DSP2_I0_full,
ega_Explode_windowa_o_adap_o_5_DSP2_I0_empty,
ega_Explode_windowa_o_adap_o_5_DSP2_I0_full,
ega_Explode_coefs_o_adap_o_5_DSP2_I0_empty,
ega_Explode_coefs_o_adap_o_5_DSP2_I0_full,
ega_Explode_windowa_o_adap_o_7_DSP2_I0_empty,
ega_Explode_windowa_o_adap_o_7_DSP2_I0_full,
ega_mul_o_DSP2_I0_empty,
ega_mul_o_DSP2_I0_full,
ega_Explode_coefs_o_adap_o_7_DSP2_I0_empty,
ega_Explode_coefs_o_adap_o_7_DSP2_I0_full,
ega_sub_o_DSP2_I0_empty,
ega_sub_o_DSP2_I0_full,
ega_filt_o_DSP2_I0_empty,
ega_filt_o_DSP2_I0_full,
ega_gensig_o_DSP2_I0_empty,
ega_gensig_o_DSP2_I0_full,
)

main_
spawn_thread(I0)
float_cst(9,1,-1,2,-5,5,8,-4,9,0,ega_Coeff_DSP1_o)
float_input(1,ega_gensig_o)
float_window(1,9,ega_gensig_o,ega_window_o)
dot_product(9,ega_window_o,ega_Coeff_DSP1_o,ega_filt_o)
float_output(1,ega_sub_o)
adaptator(ega_mul_o,ega_Explode_windowa_o_adap_o_7,ega_Explode_coefs_o_adap_o_7,ega_adap_7_o)
adaptator(ega_mul_o,ega_Explode_windowa_o_adap_o_5,ega_Explode_coefs_o_adap_o_5,ega_adap_5_o)
adaptator(ega_mul_o,ega_Explode_windowa_o_adap_o_3,ega_Explode_coefs_o_adap_o_3,ega_adap_3_o)
adaptator(ega_mul_o,ega_Explode_windowa_o_adap_o_1,ega_Explode_coefs_o_adap_o_1,ega_adap_1_o)
Pre1_(ega_sub_o_DSP1_I0_empty,I0)
Pre1_(ega_Explode_windowa_o_adap_o_7_DSP1_I0_empty,I0)
Pre1_(ega_mul_o_DSP1_I0_empty,I0)
Pre1_(ega_Explode_coefs_o_adap_o_7_DSP1_I0_empty,I0)
Pre1_(ega_Explode_windowa_o_adap_o_5_DSP1_I0_empty,I0)
Pre1_(ega_Explode_coefs_o_adap_o_5_DSP1_I0_empty,I0)
Pre1_(ega_Explode_windowa_o_adap_o_3_DSP1_I0_empty,I0)
Pre1_(ega_Explode_coefs_o_adap_o_3_DSP1_I0_empty,I0)
Pre1_(ega_Explode_windowa_o_adap_o_1_DSP1_I0_empty,I0)
Pre1_(ega_Explode_coefs_o_adap_o_1_DSP1_I0_empty,I0)

endloop_
saveFrom_(,DSP2)
endthread_

SynDEx-6.6.2 : C:/Program Files/syndex-6.6.2/examples/ega/ega2/DSP2.m4
Pre0_(ega_Explode_windowa_o_adap_o_1_DSP1_I0_full)
Suc1_(ega_adap_7_o_DSP1_I0_full)
send_(ega_adap_7_o_C6x,DSP1,DSP2)
Pre0_(ega_adap_7_o_DSP1_I0_empty)
Suc1_(ega_adap_5_o_DSP1_I0_full)
send_(ega_adap_5_o_C6x,DSP1,DSP2)
Pre0_(ega_adap_5_o_DSP1_I0_empty)
Suc1_(ega_adap_3_o_DSP1_I0_full)
send_(ega_adap_3_o_C6x,DSP1,DSP2)
Pre0_(ega_adap_3_o_DSP1_I0_empty)
Suc1_(ega_adap_1_o_DSP1_I0_full)
send_(ega_adap_1_o_C6x,DSP1,DSP2)
Pre0_(ega_adap_1_o_DSP1_I0_empty)
endloop_
saveFrom_(,DSP2)
endthread_

main_
spawn_thread(I0)
float_cst(9,1,-1,2,-5,5,8,-4,9,0,ega_Coeff_DSP1_o)
float_input(1,ega_gensig_o)
float_window(1,9,ega_gensig_o,ega_window_o)
dot_product(9,ega_window_o,ega_Coeff_DSP1_o,ega_filt_o)
float_output(1,ega_sub_o)
adaptator(ega_mul_o,ega_Explode_windowa_o_adap_o_7,ega_Explode_coefs_o_adap_o_7,ega_adap_7_o)
adaptator(ega_mul_o,ega_Explode_windowa_o_adap_o_5,ega_Explode_coefs_o_adap_o_5,ega_adap_5_o)
adaptator(ega_mul_o,ega_Explode_windowa_o_adap_o_3,ega_Explode_coefs_o_adap_o_3,ega_adap_3_o)
adaptator(ega_mul_o,ega_Explode_windowa_o_adap_o_1,ega_Explode_coefs_o_adap_o_1,ega_adap_1_o)
Pre1_(ega_sub_o_DSP1_I0_empty,I0)
Pre1_(ega_Explode_windowa_o_adap_o_7_DSP1_I0_empty,I0)
Pre1_(ega_mul_o_DSP1_I0_empty,I0)
Pre1_(ega_Explode_coefs_o_adap_o_7_DSP1_I0_empty,I0)
Pre1_(ega_Explode_windowa_o_adap_o_5_DSP1_I0_empty,I0)
Pre1_(ega_Explode_coefs_o_adap_o_5_DSP1_I0_empty,I0)
Pre1_(ega_Explode_windowa_o_adap_o_3_DSP1_I0_empty,I0)
Pre1_(ega_Explode_coefs_o_adap_o_3_DSP1_I0_empty,I0)
Pre1_(ega_Explode_windowa_o_adap_o_1_DSP1_I0_empty,I0)
Pre1_(ega_Explode_coefs_o_adap_o_1_DSP1_I0_empty,I0)
```



Génération d'exécutif



- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- Optimisation
- Génér. code
- Conclusion

Allocation mémoire

```

include(syndx.m4x)dnl
processor_(U,root, ega2u,SynDEx v5, 2001)

semaphores_(
  sub_o_empty_L0_root, sub_o_full_L0_root,
  filt_o_empty_L0, filt_o_full_L0,
  gensig_o_empty_L0_p, gensig_o_full_L0_p,
)
alloc_(float,sub_o)
alloc_(float,gensig_o)
alloc_(float,wind_o, 9)
alloc_(float,coef1_a, 9)
alloc_(float,fil_o)

thread_(TCP,L0, root, p)
loadDnto_(, p)
Pre0_(gensig_o_empty_L0_p)
Pre0_(filt_o_empty_L0)
loop_
  Suc1_(gensig_o_full_L0_p)
  send_(gensig_o, U,root,p)
  Pre0_(gensig_o_empty_L0_p)
  Suc1_(filt_o_full_L0)
  send_(filt_o, U,root,p)
  Pre0_(filt_o_empty_L0)
  Suc1_(sub_o_empty_L0_root)
  recv_(sub_o, U,p,root)
  Pre0_(sub_o_full_L0_root)
endloop_
endthread_

```

Sequence de communication

Sequence de calcul

```

main_
spawn_thread_(L0)
gensig()
wind(wind_o)
coef1(coef1_a)
visu()
Pre1_(sub_o_empty_L0_root)
loop_
  Suc0_(gensig_o_empty_L0_p)
  gensig(gensig_o)
  Pre1_(gensig_o_full_L0_p)
  shift_(wind_o, gensig_o)
  Suc0_(filt_o_empty_L0)
  gdotProd(coef1_a, wind_o, ...)
  Pre1_(filt_o_full_L0)
  Suc0_(sub_o_full_L0_root)
  visu(sub_o)
  Pre1_(sub_o_empty_L0_root)
endloop_
visu()
wind(wind_o)
gensig()
wait_endthread_(L0)
endmain_

endprocessor_

```

Opérations «gensig»

Opérations «gdotProd»

Emission

reception

synchro

synchro



Association macro \Leftrightarrow fnct C



nomAppli.m4x

Fonction Reduction

divert

```
void reduction(int height, unsigned char *im, int width, unsigned short *imreduc);
```

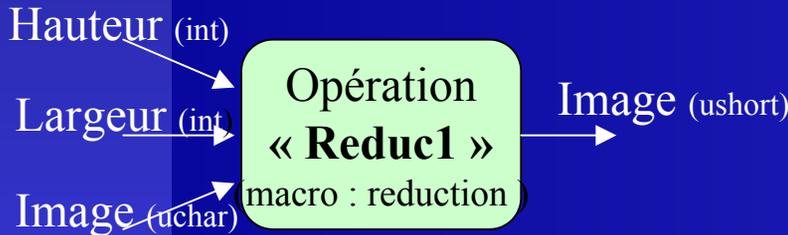
divert(-1)

```
def ('reduction', `ifelse(MGC,LOOP, `proto_(int?$1, uchar[65536]?$3, int?$2, ushort[16384]!$4)
    Ccall_(void, `reduction', int $1, uchar *$3, int $2, ushort *$4)')')
```



- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- Optimisation
- Génér. code
- Conclusion

reduction(cst_h, cst_L, capt_o, reduc1_o)



```

void reduction(int height, unsigned char *im, int width,
unsigned short *imreduc)
{
  int i, j, k;
  for (j = 0; j < height; j++)
  {
    for (i = 0; i < width; i++)
    {
      k=(j >> 1)*(width >> 1)+(i >> 1);
      imreduc[k] += im[j*width+i];
    }
  }
}
  
```

fonctions.c

(Ccall et proto : définition dans syndex.m4x et c.m4x)



Conclusion



- Contexte
- AAA/SynDEx
- Architecture
- Algorithme
- Implantation
- Optimisation
- Génér. code
- **Conclusion**

- **Extension de AAA aux circuits reconfigurables :**
 - **thèses ESIEE : A. Dias (achevée), L. Kaouane (en cours)**
 - **Développement à l'ESIEE de la version SynDEx-IC (disponible à la rentrée)**
- **Extension aux architectures mixtes FPGA-DSP :**
 - **Co-design, SoC (Excalibur, Virtex Pro II..)**
 - **Prochaine thèse ESIEE en septembre**

