Processor Architecture Laboratory
Laboratoire d'Architecture des Processeurs

School of Computer and Communication Sciences

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Développement embarqué XScale

Projet de diplôme

Julien Pilet
Stéphane Magnenat

Responsable : René Beuchat
Professeur : Paolo Ienne

14 mars 2003

Plan de la présentation

- Introduction
- Directions et choix
- Outils de développement
- OS : Linux
- Système multimédia embarqué
- Consommation et performances
- Conclusion
- Démonstration

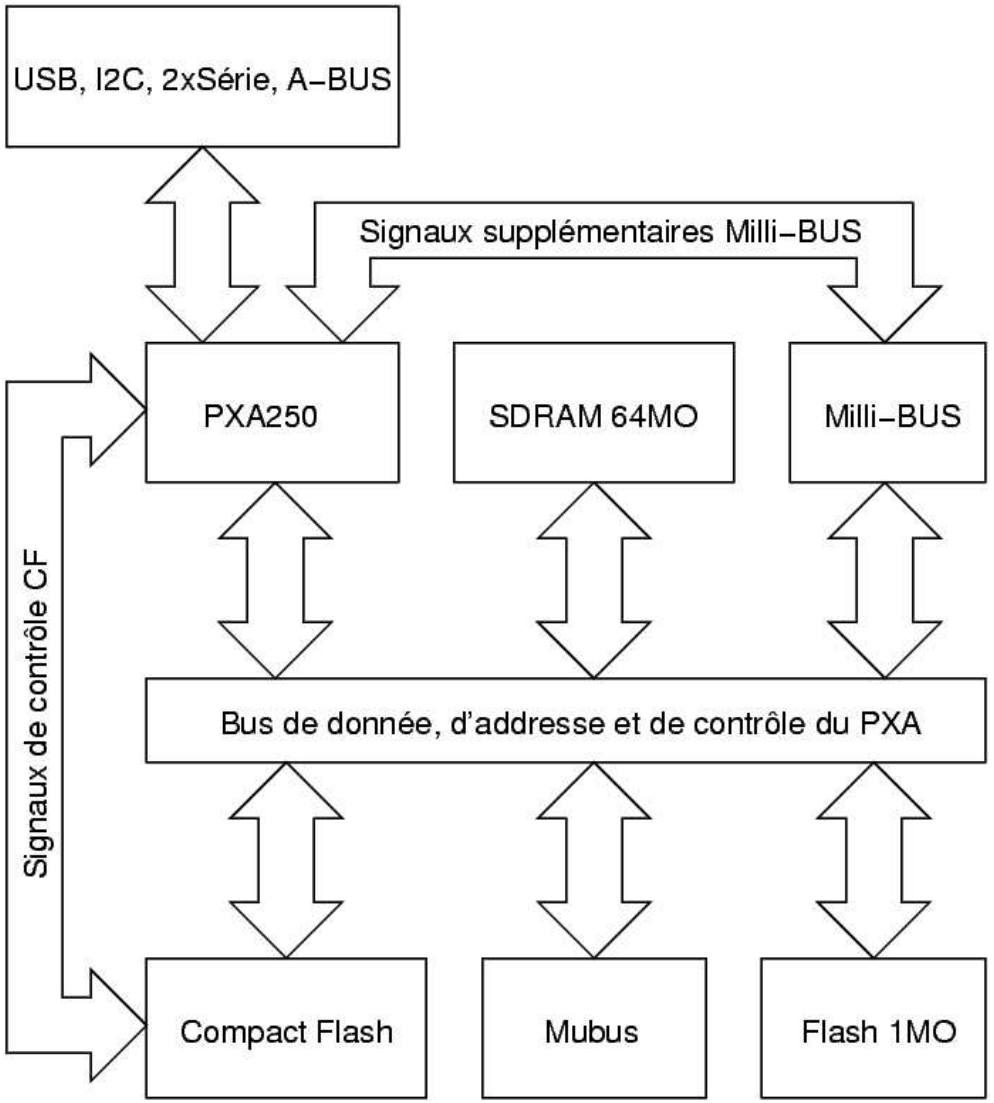
Introduction : Objectifs

- Suite de notre projet de semestre
- Mettre en valeur Armonie :
  - Support pédagogique
  - Système multimédia embarqué
  - Pilotage robot Cyclope

Introduction : Matériel de base

- Armonie
  - processeur ARM XScale PXA250/PXA255
  - 64 MB SDRAM
  - 1 MB Flash
  - CompactFlash
  - USB, Série, I²C, JTAG, Mubus
  - Extensions (Milli-BUS)
  - Alimentation séparée (A-BUS)

Matériel de base : Armonie



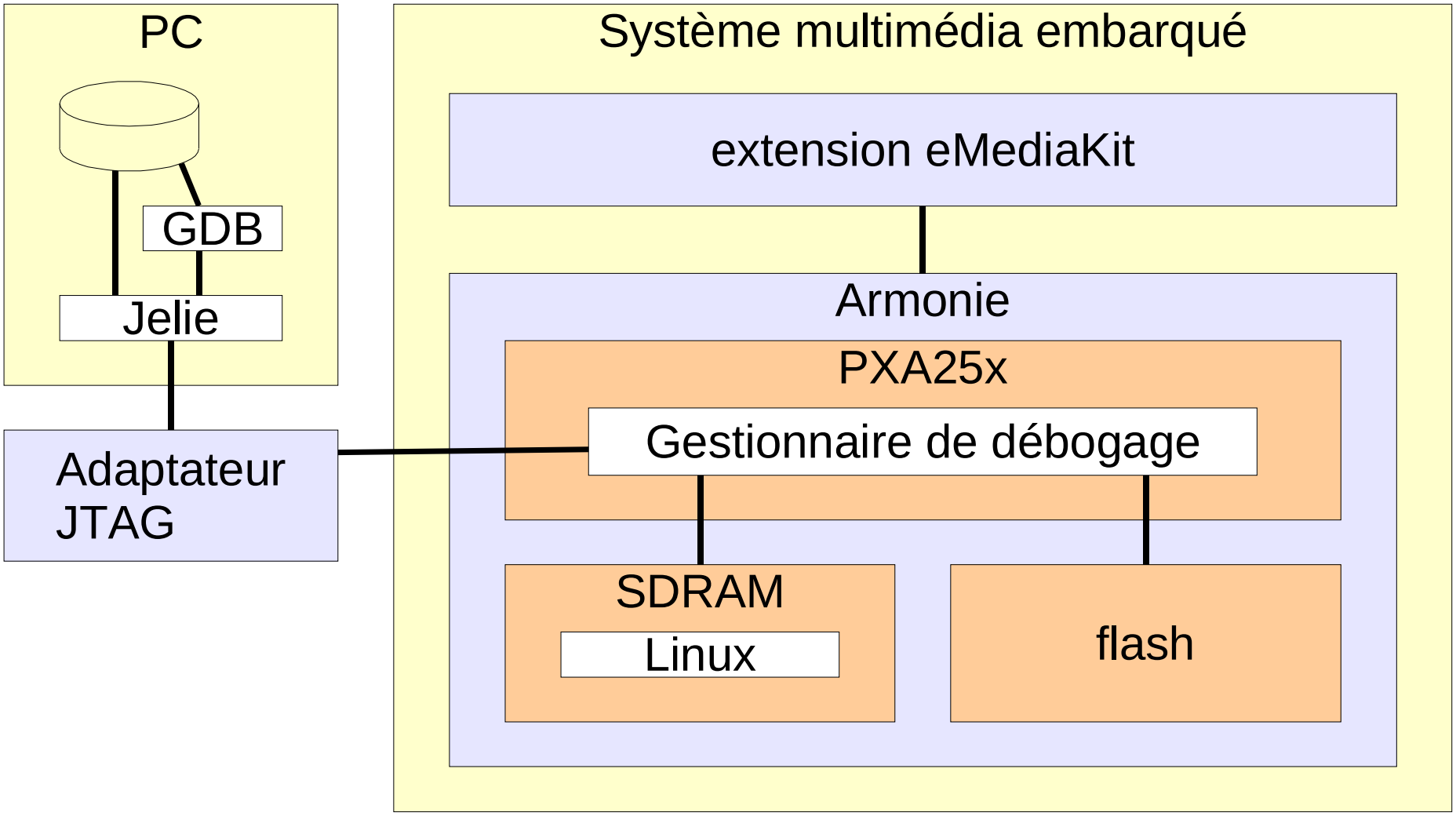
Introduction : A propos du PXA25x

- Processeur ARM
  - architecture v5
  - instructions RISC & DSP-like
  - support Thumb
  - fréquences et tensions variables
- 68kB de caches mémoire :
  - 2 x 32kB caches de données et instructions
  - 2 x 2kB de mini-caches de données et instructions

Introduction : A propos du PXA25x

- Périphériques intégrés :
  - contrôleurs SDRAM, PCMCIA/CF, LCD
  - IO : GPIO, série, I²C, I²S, SSP, AC97, USB (esclave), MMC
- 2 tensions d'alimentation :
  - 3.3V : bus et GPIO
  - variable 0.85-1.3V : coeur PXA25X
- Fréquences :
  - bus interne : 50 MHz - 166 MHz
  - coeur : 100 MHz - 400 MHz

Introduction : Vue générale



Choix

- Support pédagogique
  - cours *Systèmes embarqués* de R. Beuchat
  - futurs projets semestre/diplôme
- Système multimédia embarqué
  - utilise Milli-BUS
- Pilotage robot Cyclope
  - possible à travers A-BUS (carte d'Adrian Spycher)

Pourquoi un système d'exploitation (OS) ?

- Développement sans OS possible, mais :
  - pas de gestion de la mémoire
  - un seul programme
  - tout est à faire *à la main*
- Le système d'expoitation fournit :
  - gestion et protection de la mémoire
  - multitâche
  - bibliothèques standards (libc + posix)
  - pilotes pour les périphériques

Choix de l'OS, comparatif

Seuls les OS supportants le PXA25x ont été pris en compte.

Problématique	**Linux**	**eCos**	**NetBSD**	**WinCE**
Support des périphériques du PXA	+ +	-	+	+ +
Communauté (développement et utilisation)	+ +	+	+	+
Temps réel	-	+ +	- -	- -
Réseau	+ +	+	+ +	+
Orienté graphique (GUI disponibles)	+ +	-	+	+ +
UNIX (multitâche, protection mémoire, etc...)	+ +	- -	+ +	-
Accès aux sources et redistribuable	+	+	+	-

Choix de l'OS : Linux

- Avantages :
  - existe, bon support pour l'ARM et le PXA25x
  - très activement développé
  - supporte les fonctionnalités d'un UNIX
  - pilotes pour pratiquement tout
- Inconvéniants :
  - documentation disparate
  - gros
  - pas de support temps réel natif
  - API non unifiée

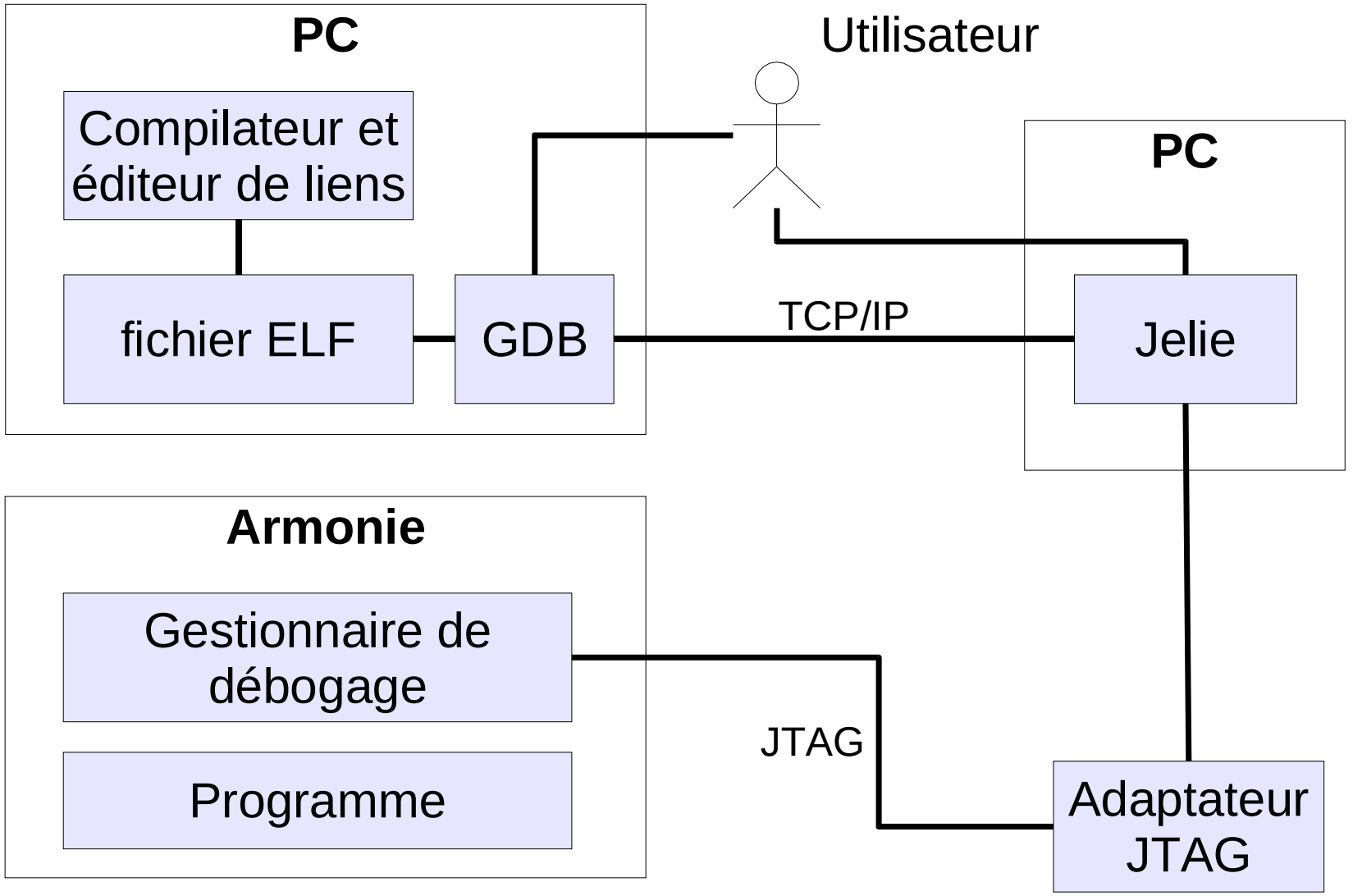
Choix du libre

- Permet une maîtrise totale grâce à l'accès aux sources
- Librement redistribuable
- Système matériel et logiciel entièrement libre
- Communauté dynamique
- Intéressant à développer
- Gratuit
- Distribué sur http://lapwww.epfl.ch/dev/arm/

Collaboration extérieure

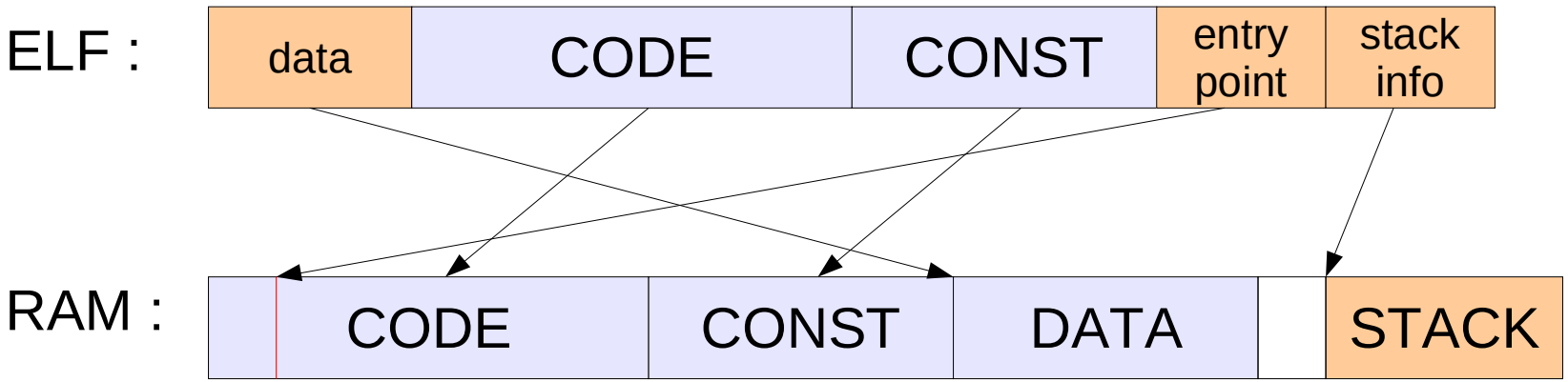
- Collaboration avec K-Team (P. Bureau) et LSA 2 (F. Mondada)
- Nos avantages :
  - augmente les chances de réussite
  - contacts industriels
  - test de nos outils
- Leurs avantages :
  - réduit les coûts
  - support gratuit

Outils de développement, schéma général

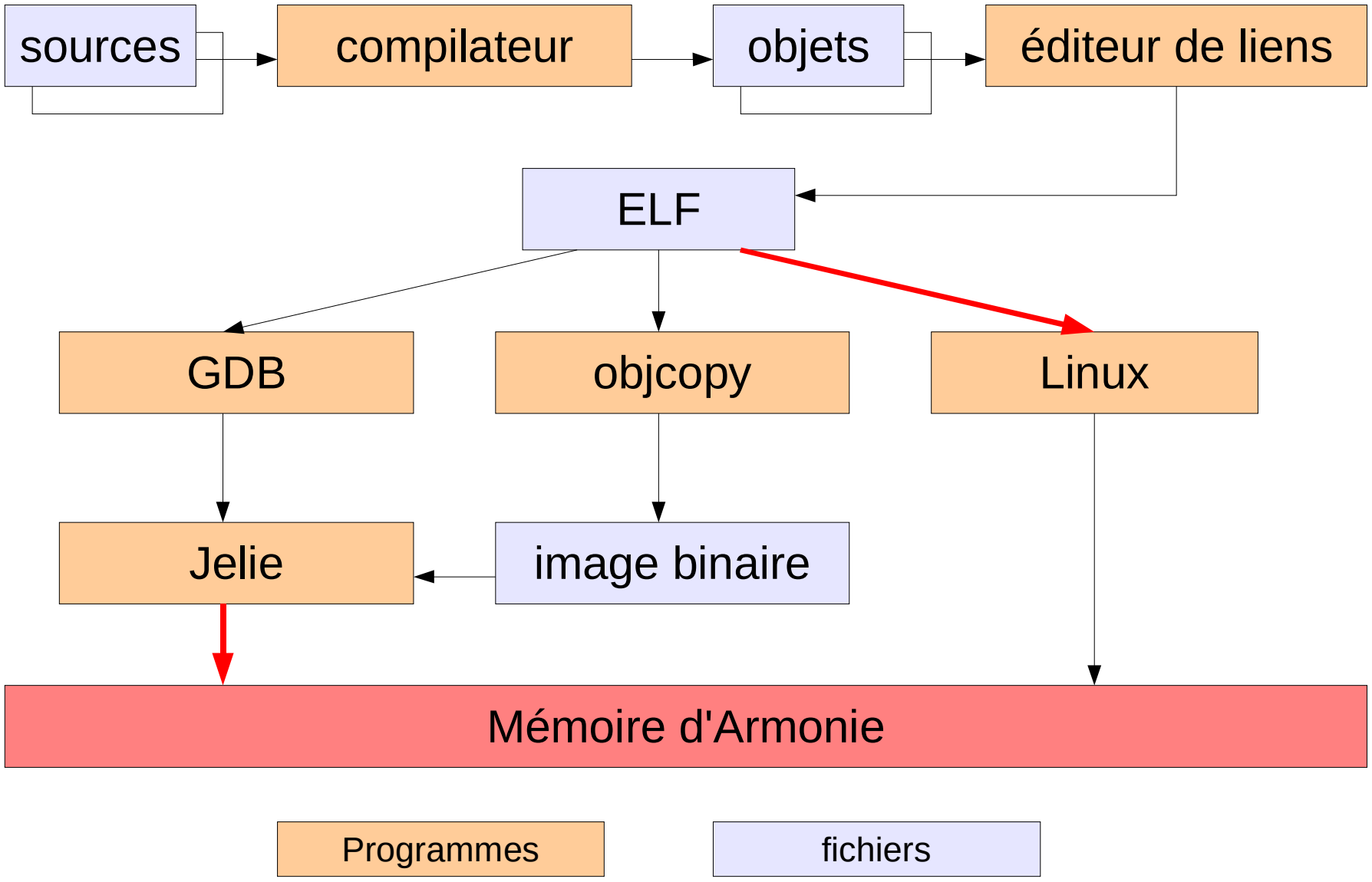


Chargement d'un programme

- Charger le binaires en mémoire, aux bonnes adresses
- Avoir une pile valide
- Sauter à la bonne adresse de départ



Chargement d'un programme



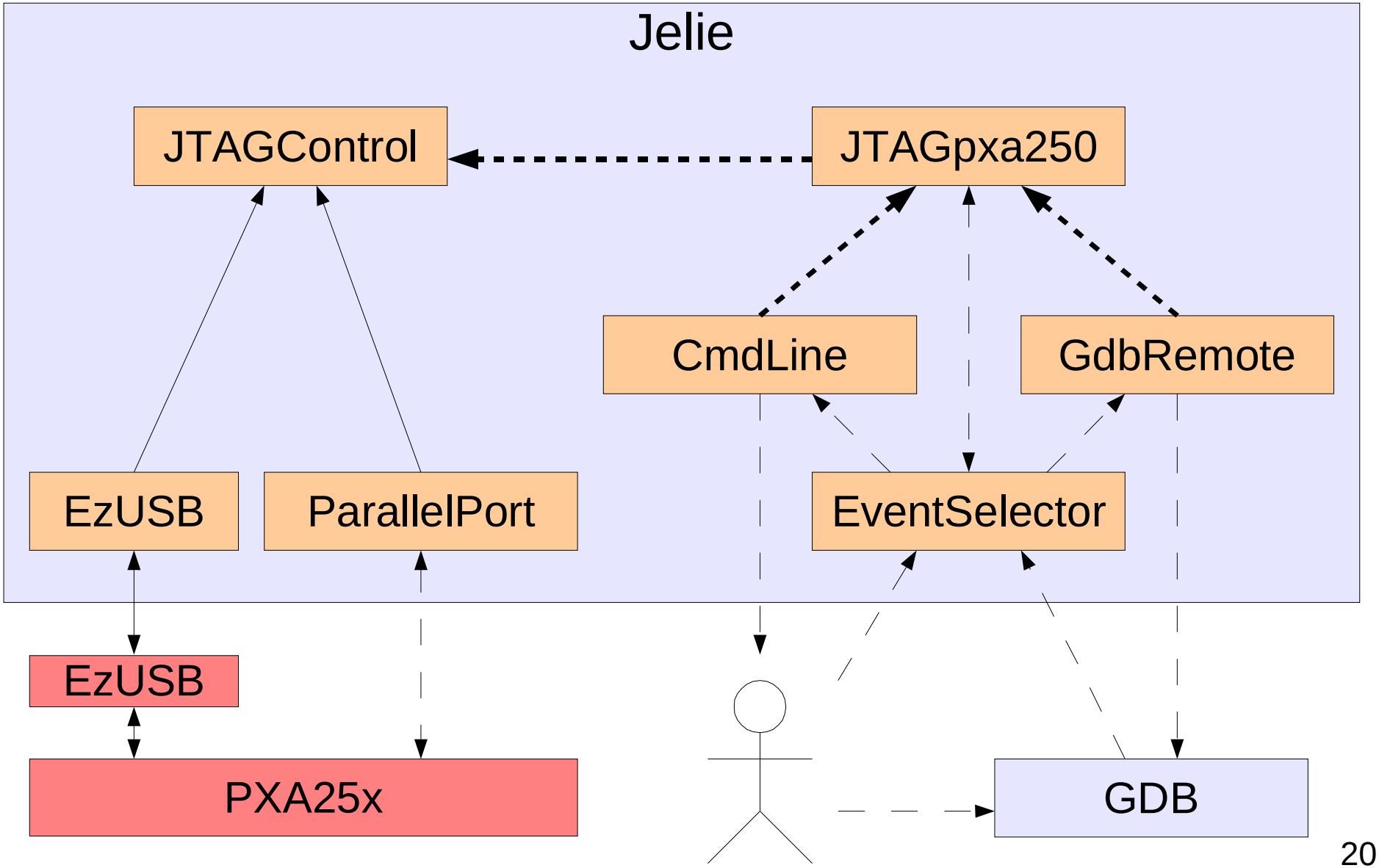
Jelie

- Pilotage JTAG
- Orienté objet
- 10000 lignes de C++ et assembleur ARM
- Charge les programmes
- Lit et écrit les registres
- Lit et écrit la mémoire
- Place des points d'arrêts
- Interface GDB
- Programme la mémoire flash

Jelie : connection JTAG physique

- JTAG : Joint Test Action Group
  - initialement prévu pour le test des circuits
  - étendu pour usages divers
- USB :
  - EzUSB
- Port parallèle :
  - Wiggler
  - Alteraprog

Jelie : schéma interne



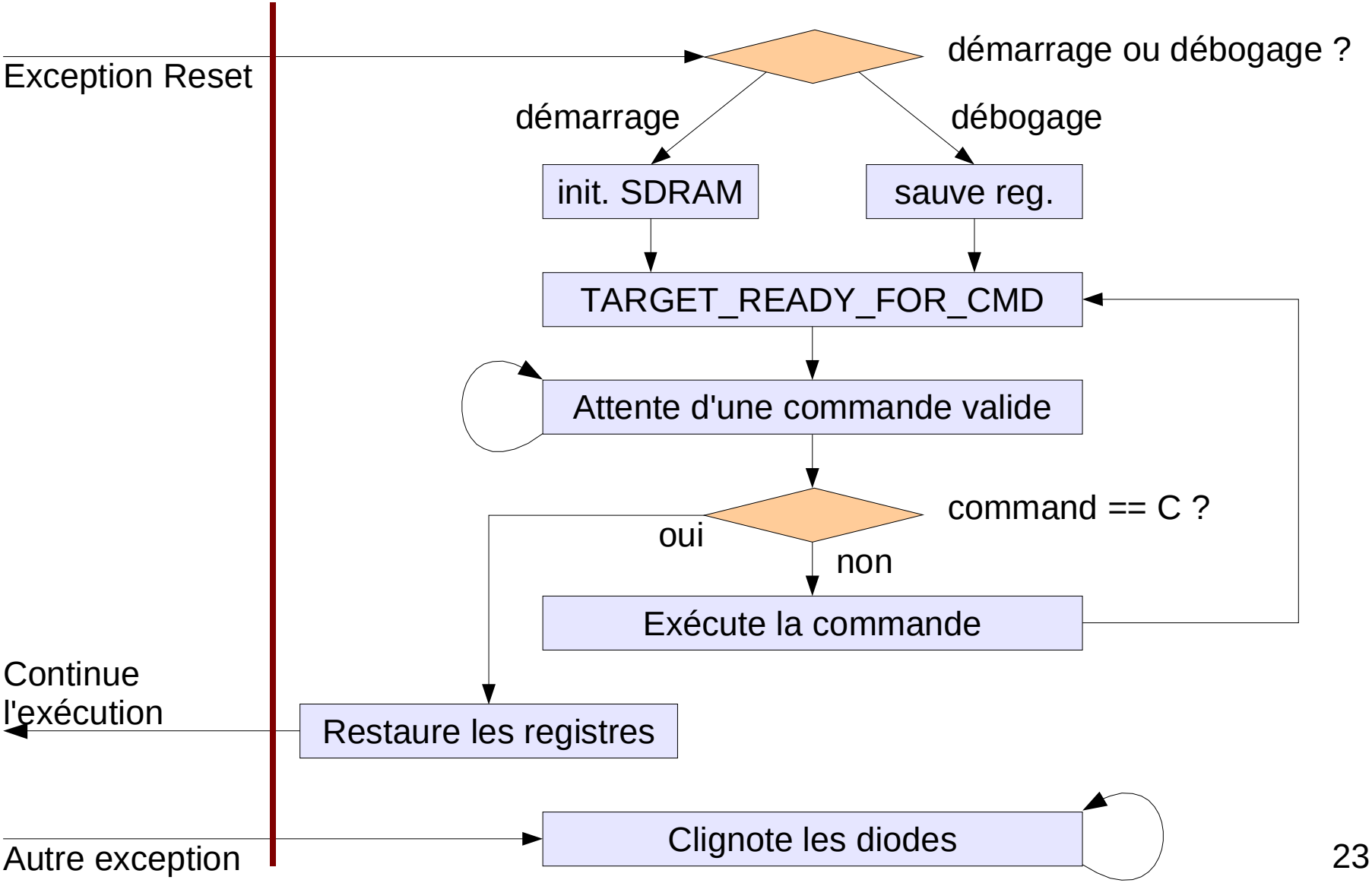
Jelie : méchanisme de débogage du PXA25x

- Par port JTAG
- Mini-cache d'instructions pour gestionnaire de débogage
- Transferts de données entre gestionnaire de débogage et PC
- 2 points d'arrêt matériels
- Aucun pas à pas matériel

Jelie : gestionnaire de débogage

- Chargé en mini-cache d'instructions par JTAG
- Assembleur ARM
- N'utilise pas la mémoire (pas de pile)
- Sauvegarde les registres sur le PC
- Appelé par :
  - démarrage
  - instruction bkpt
  - point d'arrêt matériel
  - événement externe JTAG

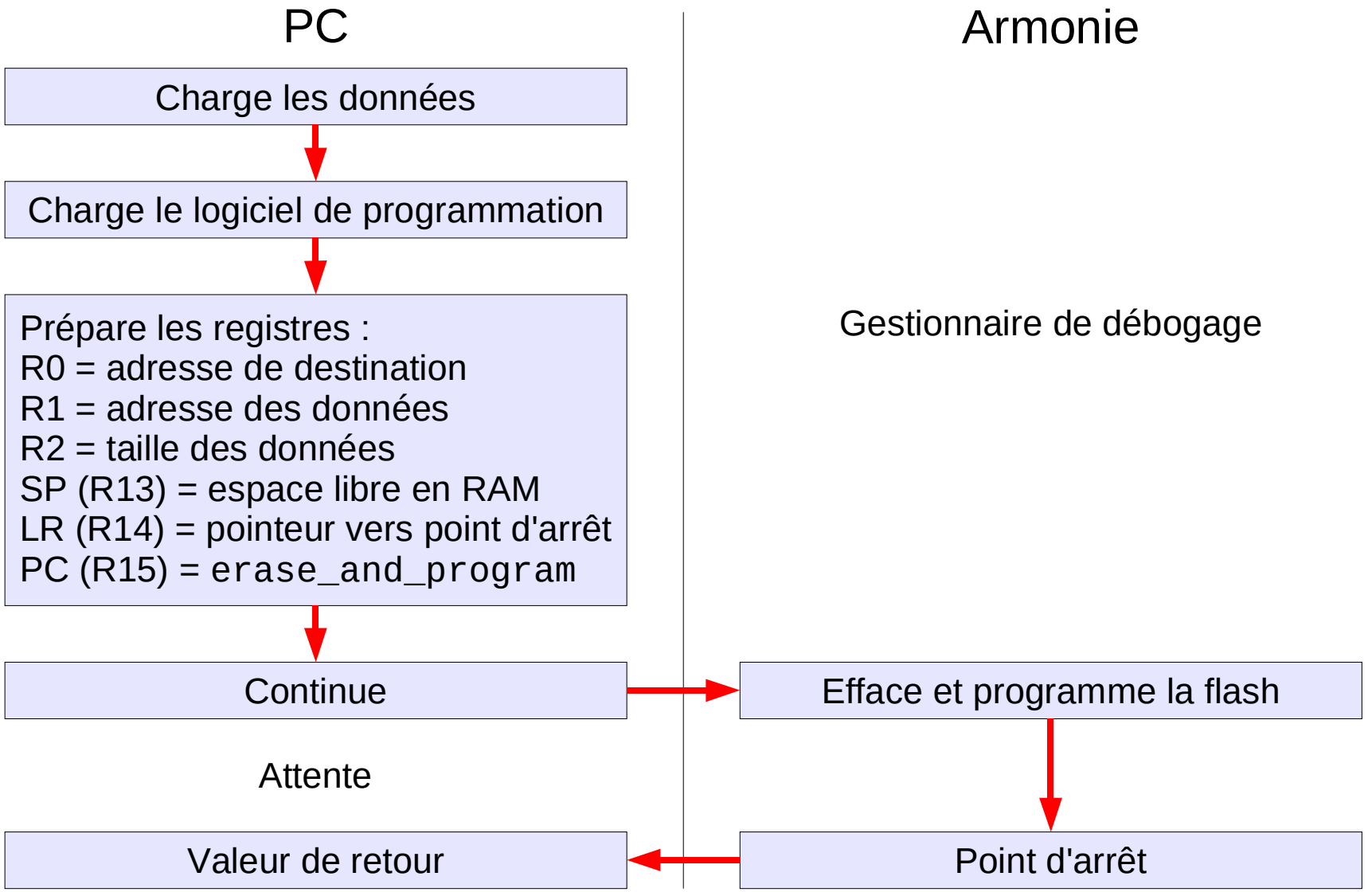
Gestionnaire de débogage : diagramme de fonctionnement



Jelie : chargement et exécution de programmes

1. Chargement en mini-cache d'instructions du gestionnaire de débogage par JTAG
2. Chargement en mémoire depuis le PC du code et des données par JTAG
3. Mise à jour des registres :
   - SP (R12) pointe sur la pile
   - PC (R15) pointe sur l'adresse à exécuter
4. Sort du mode de débogage et continue l'exécution

Jelie : mémoire Flash et appels de fonctions embarquées



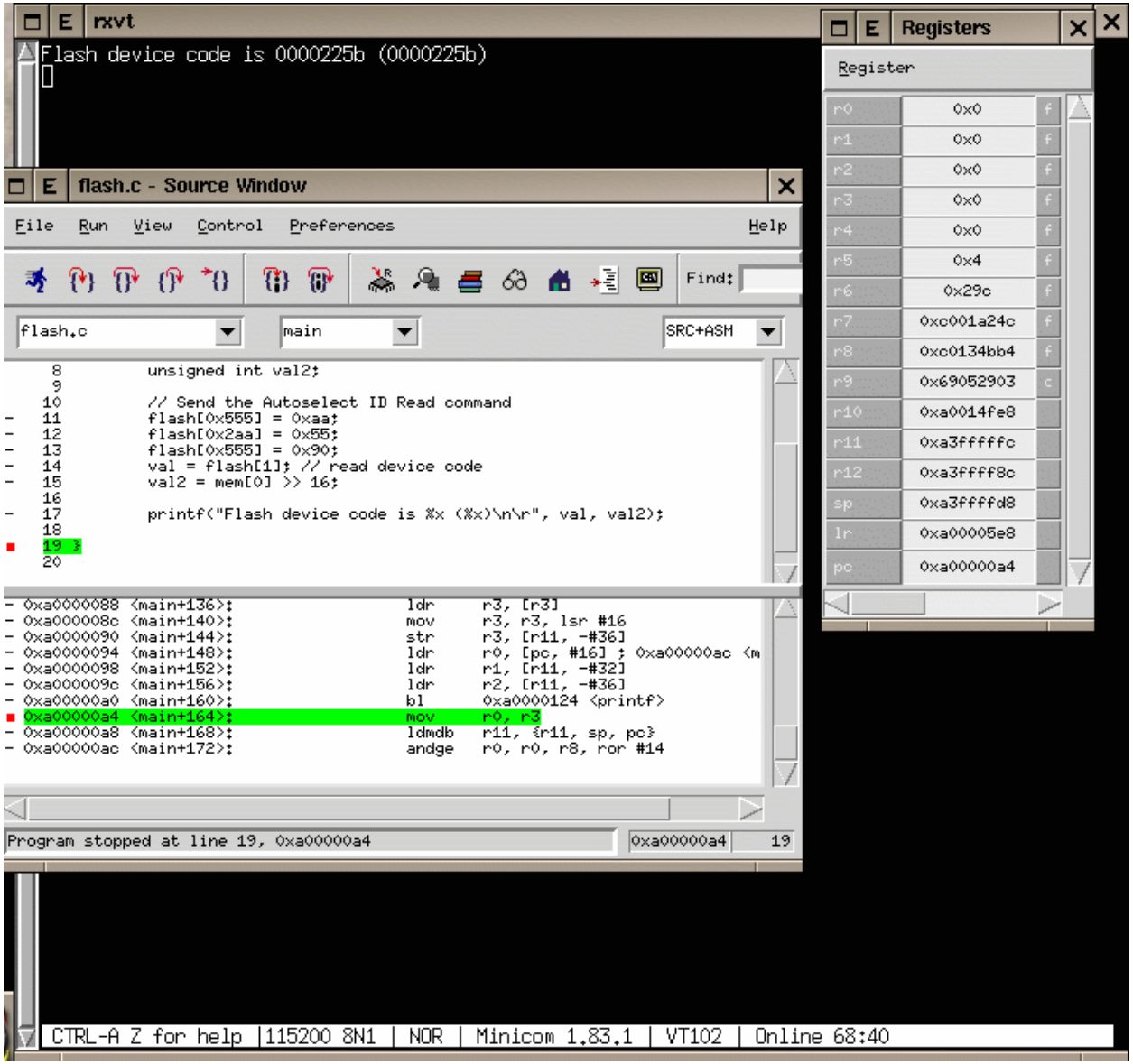
Jelie : mémoire Flash et appels de fonctions embarquées

```
load flash/flash.bin    # load the remote program
load \1 a1000000        # load data to write
register 0 \2           # set destination address
register 1 a1000000     # where the data is stored
register 2 \3           # number of words to write
register 13 a0c00000    # SP = 0xA0000000 + 12 MB
register 14 $_endless$  # LR = _endless
register 15 $erase_and_program$ # set PC
continue        # launch flash programation
wait            # wait until the debug handler is ready
register 0      # display result code
```

Jelie et GDB

- GDB travaille avec :
  - des fichiers ELF
  - des sources (C, C++, ASM)
- Communication TCP/IP avec Jelie
- Jelie exécute les ordres de bas niveau :
  - lire/écrire les registres
  - lire/écrire la mémoire
  - points d'arrêt matériels
  - continuer/interrompre l'exécution

Jelie et GDB


rxvt
Flash device code is 0000225b (0000225b)
flash.c - Source Window
File Run View Control Preferences Help
Find:
flash.c
main
SRC+ASM
8 unsigned int val2;
9
10 // Send the Autoselect ID Read command
11 flash[0x555] = 0xaa;
12 flash[0x2aa] = 0x55;
13 flash[0x555] = 0x90;
14 val = flash[1]; // read device code
15 val2 = mem[0] >> 16;
16
17 printf("Flash device code is %x (%x)\n\r", val, val2);
18
19 }
20
0xa0000088 <main+136>: ldr r3, [r3]
0xa000008c <main+140>: mov r3, r3, lsr #16
0xa0000090 <main+144>: str r3, [r11, -#36]
0xa0000094 <main+148>: ldr r0, [pc, #16] ; 0xa00000ac <m
0xa0000098 <main+152>: ldr r1, [r11, -#32]
0xa000009c <main+156>: ldr r2, [r11, -#36]
0xa00000a0 <main+160>: bl 0xa0000124 <printf>
0xa00000a4 <main+164>: mov r0, r3
0xa00000a8 <main+168>: ldmdb r11, {r11, sp, pc}
0xa00000ac <main+172>: andge r0, r0, r8, ror #14
Program stopped at line 19, 0xa00000a4
0xa00000a4
19
Registers
Register
r0 0x0
r1 0x0
r2 0x0
r3 0x0
r4 0x0
r5 0x4
r6 0x29c
r7 0xc001a24c
r8 0xc0134bb4
r9 0x69052903
r10 0xa0014fe8
r11 0xa3fffffc
r12 0xa3ffff8c
sp 0xa3ffffd8
lr 0xa00005e8
pc 0xa00000a4
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 1.83.1 | VT102 | Online 68:40

Jelie : changement de fréquence

- Par appel de fonction embarquée
- Changement de la fréquence du bus interne (PXBus)
  - 50 MHz
  - 100 MHz
- Passage en mode Turbo
  - 100 MHz
  - 200 MHz
  - 300 MHz et 400 MHz (certains modèles)
- Non conforme à la documentation d'Intel

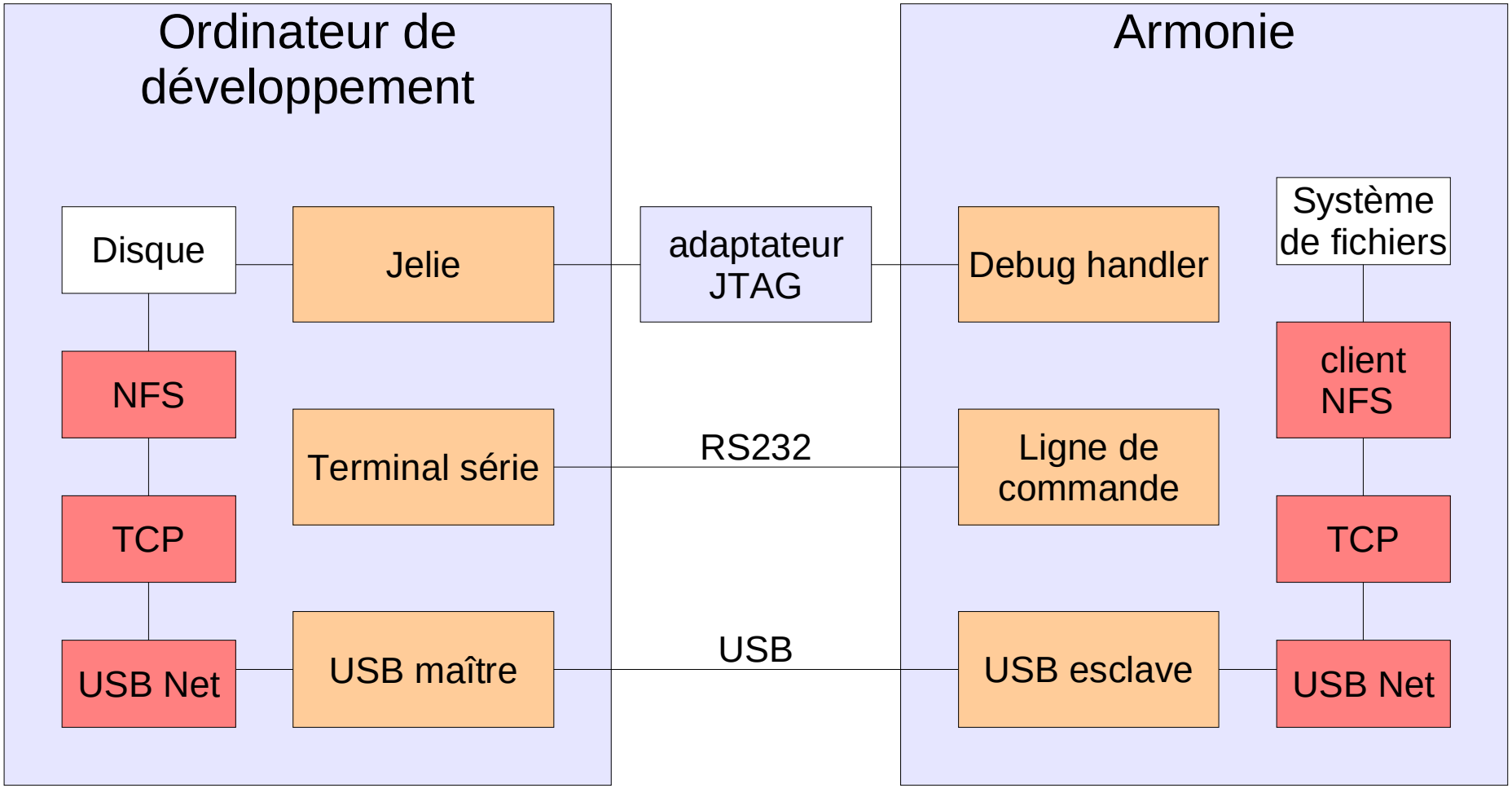
Linux sur Armonie

- Noyau 2.4.19-rmk4-pxa2-armonie1 et 2.4.19-rmk6-pxa1-armonie1
  - rmk = modifications ARM de Russell King
  - pxa = modifications PXA2xx de Nicolas Pitre (MontaVista)
  - armonie = nos modifications
- Système d'exploitation
  - BusyBox/Linux embarqué dans la mémoire flash (~900 kB)
  - GNU/Linux, distribution PsiLinux, par réseau USB: 32 MB

Noyau Linux

- ARM et microarchitecture XScale supportée
- Pilotes
  - port série
  - I²C
  - USB esclave
  - Compact Flash (adaptation commencée)
  - LCD (adapté)
  - Mubus (créé)

Linux : connections avec un ordinateur de développement



Linux : chargement depuis un ordinateur de développement

- Jelie :
  1. envoie l'image compressée du noyau en RAM,
  2. envoie l'image compressée du système de fichiers en RAM,
  3. affecte les registres,
  4. saute au début de l'image compressée du noyau.

Linux : chargement depuis la mémoire Flash

- Flallo: Flash Linux Loader
  1. copie image compressée du noyau en RAM
  2. copie système de fichier en RAM
  3. appelle l'image compressée du noyau

Linux : démarrage

- L'image compressée du noyau :
  1. décompresse le noyau à l'addresse, a0008000
  2. saute à cette addresse.
- Linux :
  - charge les pilotes de périphériques,
  - initialise la MMU, l'ordonnanceur, ...
  - décompresse le système de fichier en RAM,
  - exécute `/sbin/init` ou l'exécutable spécifié par `init=` dans la ligne de commande

Linux : fonctionnalités sur Armonie

- Multi-tâche
- Mémoire protégée
- Système de fichier en mémoire
- Bibliothèques dynamiques
- Réseau (via USB)
- Système de fichiers réseau (NFS)
- Affichage sur LCD (par DMA)
- Depuis 2.4.19-rmk6-pxa1-armonie1: changement de fréquence au vol

Linux : Système complet

- Un noyau ne suffit pas
- Programmes utilisateurs:
  - BusyBox (commandes unix en environ 400 kB)
  - PsiLinux (distribution GNU/Linux pour ARM, 32 MB)
  - Programmes autonomes avec uClibc
- Stockage :
  - compressé en flash, puis en mémoire vive
  - par réseau USB

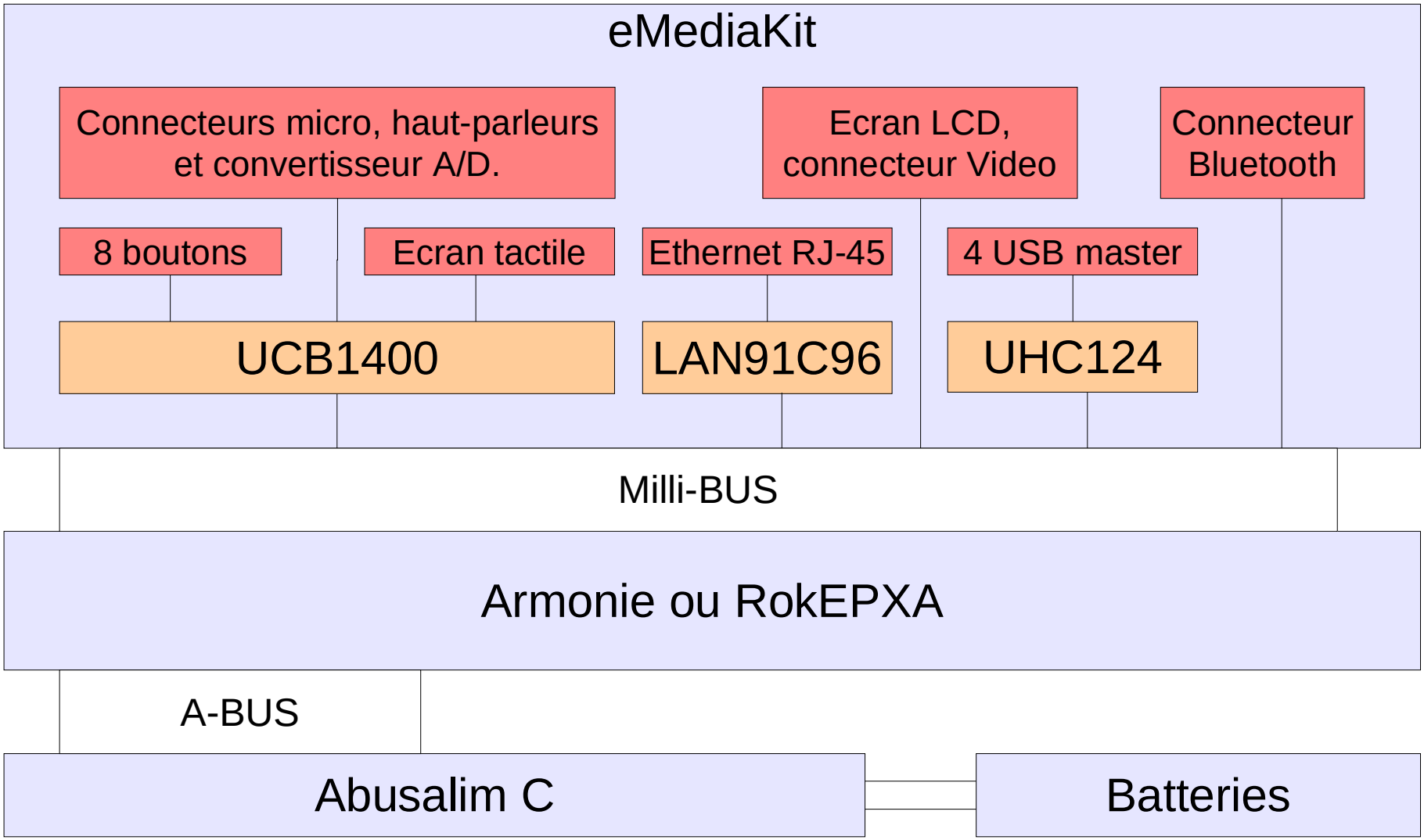
Linux comme environnement de travail

- Facilement utilisable à distance
- Automatisation aisée
- Nombreux outils de développement disponibles

Système multimédia embarqué,

- Système comprenant :
  - carte d'extension eMediaKit
  - écran LCD
  - Armonie ou RokEPXA
  - carte d'alimentation
  - batteries
- Conçu, partiellement réalisé
- En collaboration avec Cédric Gaudin

Système multimédia embarqué, schéma général



eMediaKit, UCB1400

- Circuit multi-fonction :
  - son (44 kHz, 16 bits, stéréo, in/out)
  - A/D et écran tactile (10 bits)
  - GPIO (10 pattes avec interruptions possibles)
- Connection au PXA25x par AC97 :
  - bus série et protocole, utilisé dans les PC, principalement pour le son
  - le PXA25x contient le contrôleur
- Non testé

eMediaKit, LCD

- LCD Sharp TFT Matrice active
  - 320x240x18 bit (6 bit/composante, 16 bits connectés (R: 5, G: 6, B: 5)
  - rétro-éclairage par tube fluorescent, consommation excessive (jusqu'à 3.5 W)
  - testé et fonctionne
- Ecran tactile
  - ajout, non inclu dans le LCD
  - résistif, connecté à l'UCB, A/D 10 bits (résolution théorique 1024x1024)
  - LCD fonctionne, écran tactile non testé

eMediaKit, USB maître

- UHC124 USB Host Controller,
- 4 ports USB maître,
- Detection / limitation de surcharge de courant inclues,
- Pas de pilote pour Linux pour l'instant.

Consommation et performances du PXA250

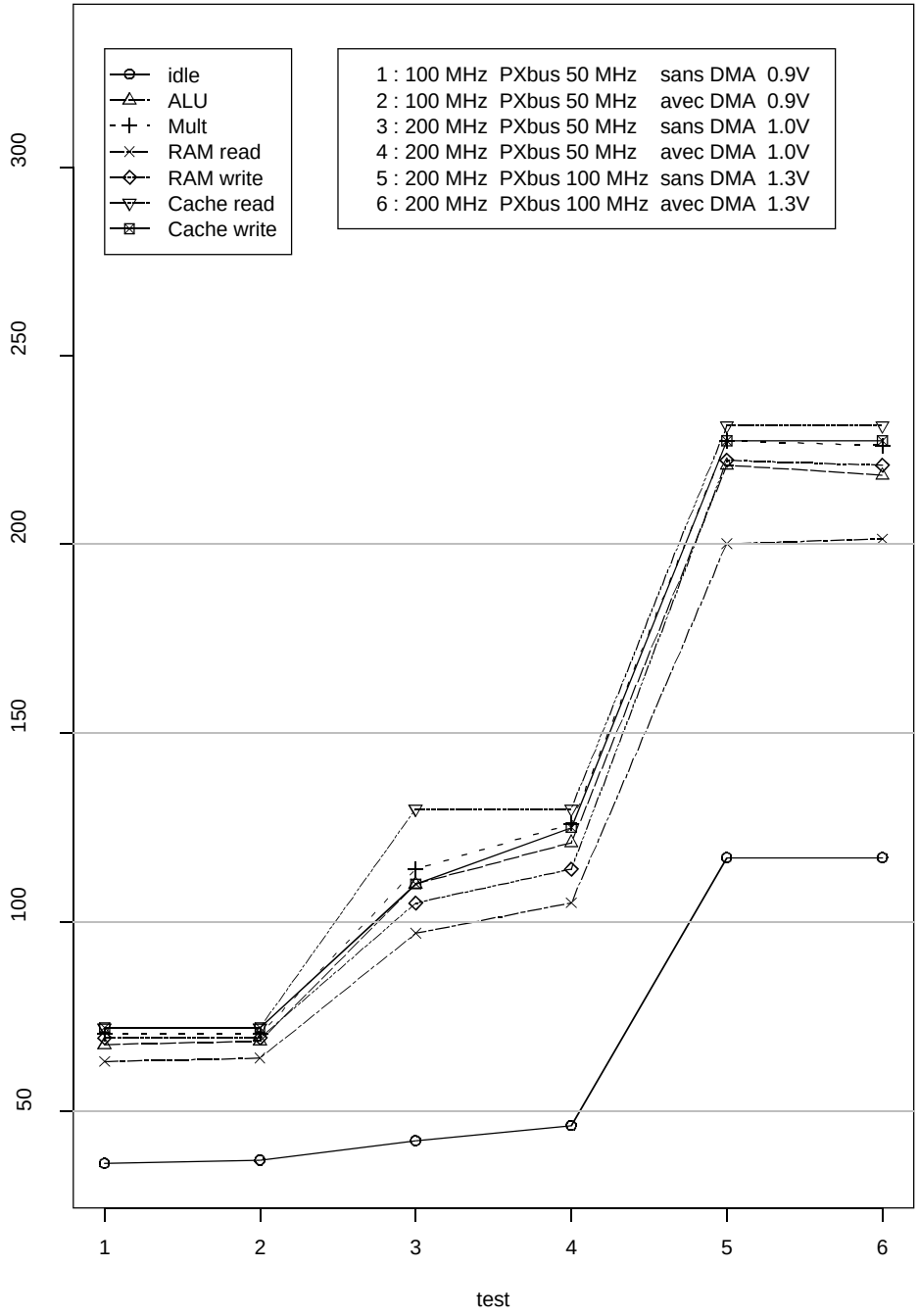
- Conditions logicielles de test :
  - inactivité
  - ALU
  - multiplications
  - lecture SDRAM
  - écriture SDRAM
  - lecture cache mémoire
  - écriture cache mémoire

Consomation

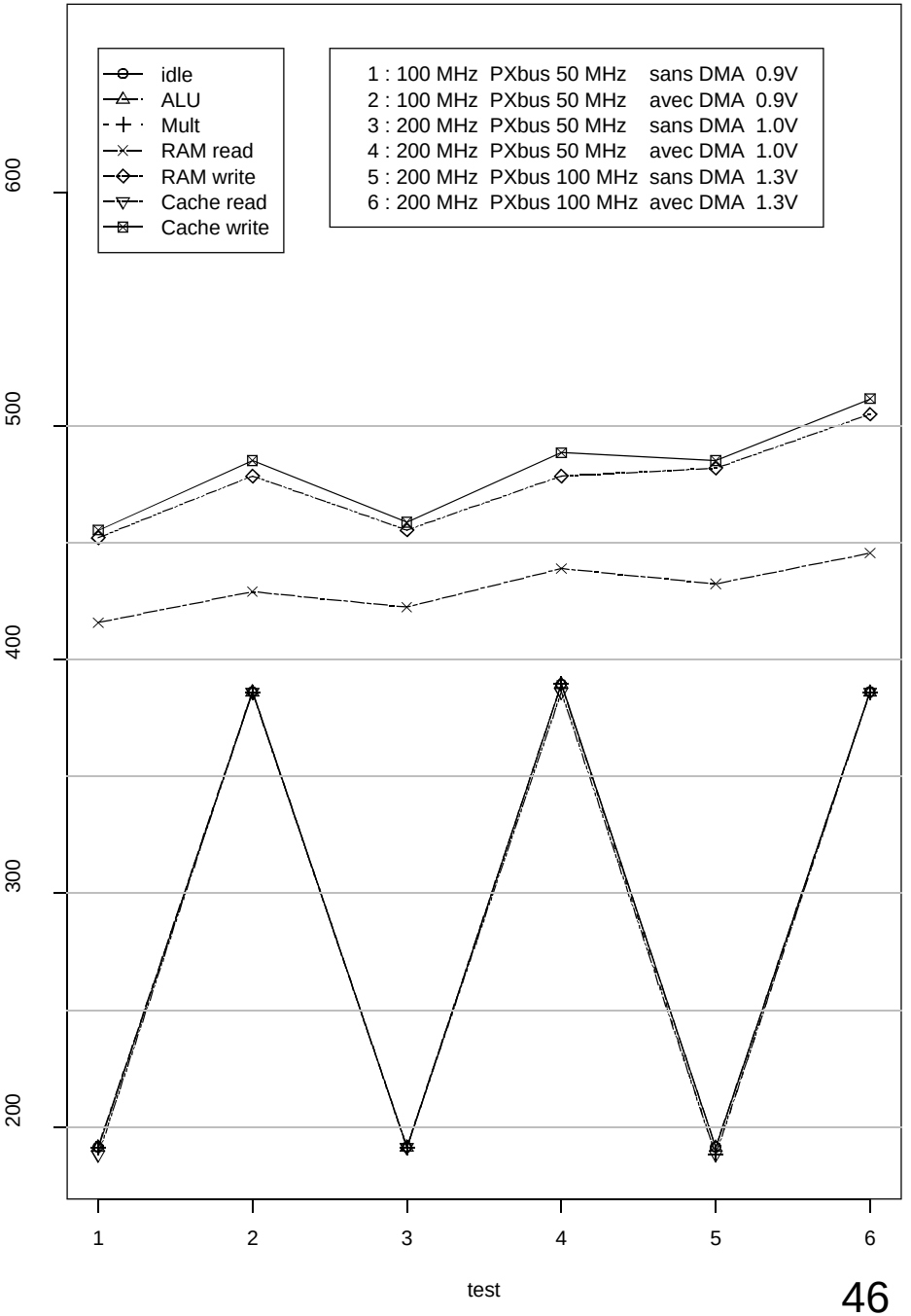
	CPU 100	CPU 200	
sans DMA	Bus 50 : 1	Bus 50 : 3	Bus 100 : 5
avec DMA	Bus 50 : 2	Bus 50 : 4	Bus 100 : 6

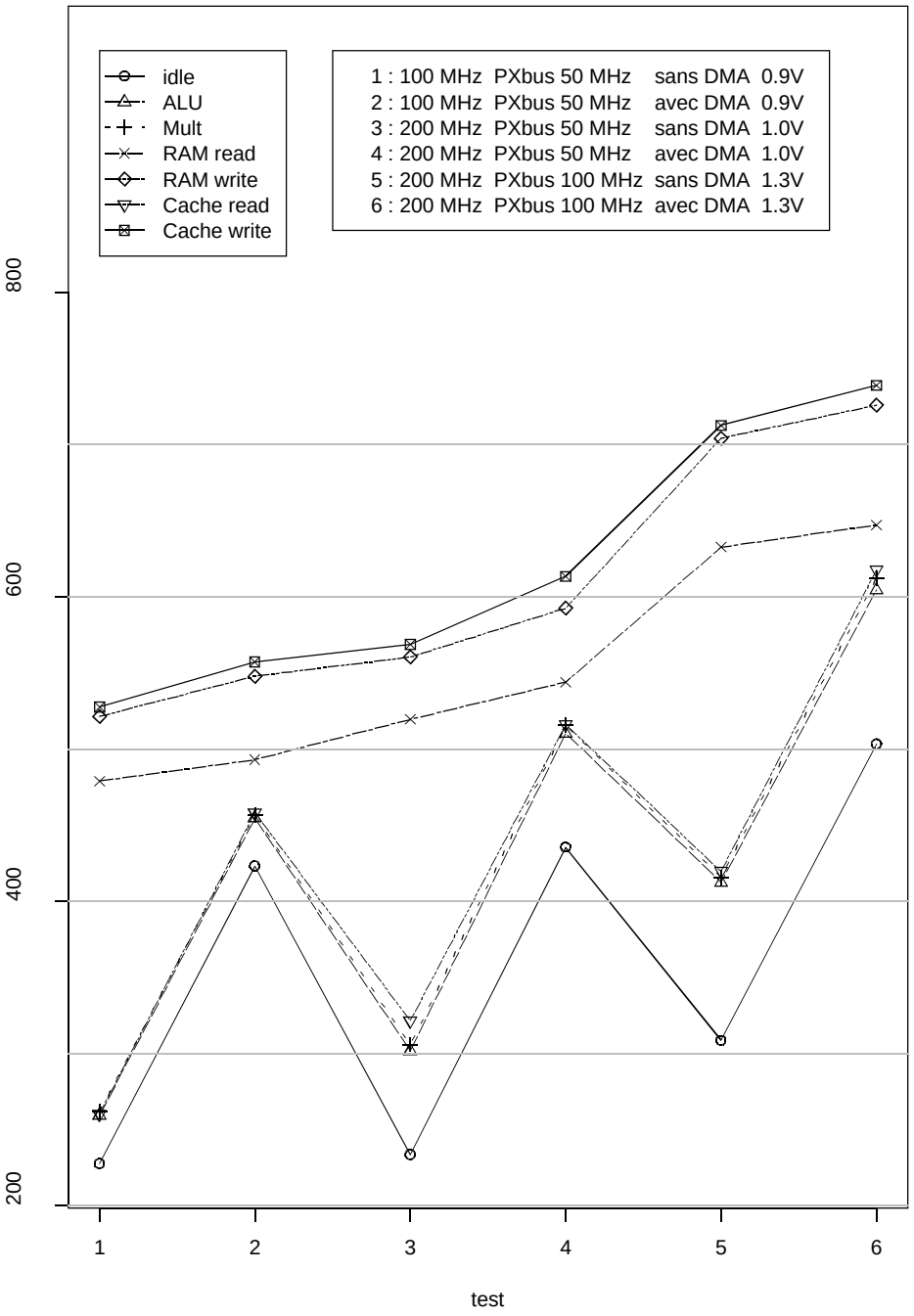
- Uniquement sur le PXA250
- Mesures :
  - courant de l'alimentation variable (0.9 - 1.3 V)
  - courant de l'alimentation 3.3 V
  - vitesse d'exécution des boucles de tests

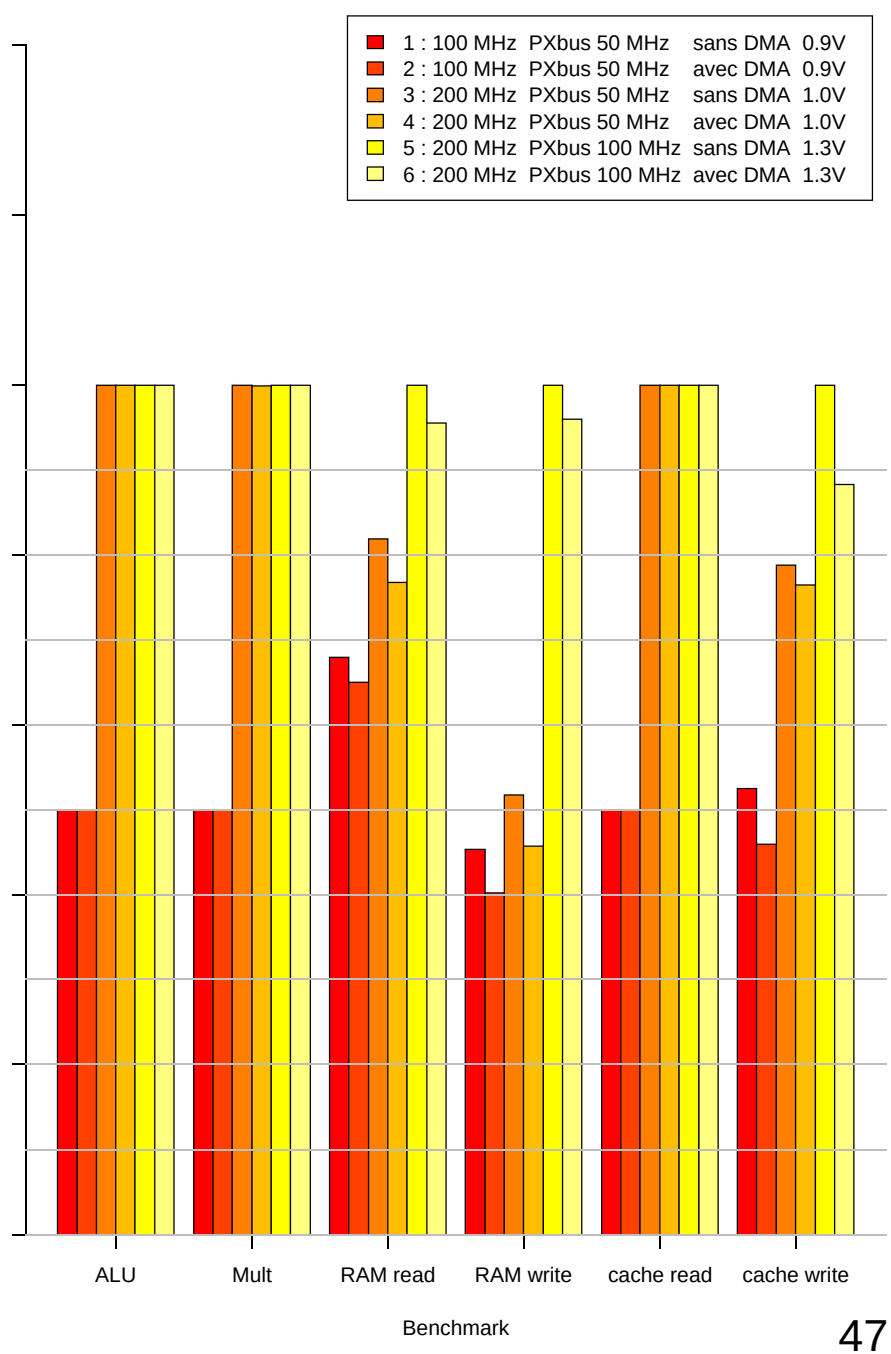
Consommation du coeur



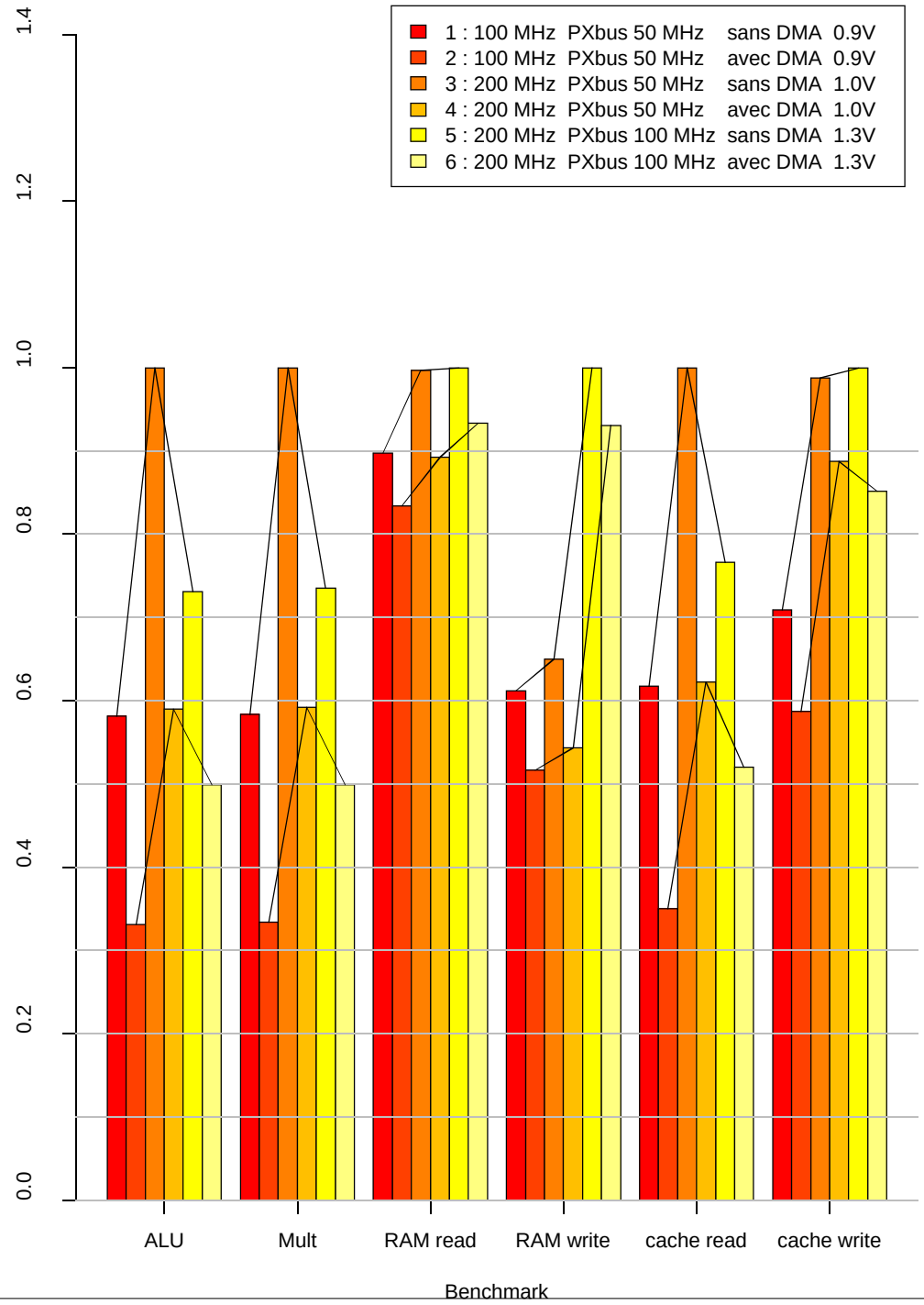
Consommation de l'alimentation 3.3V




Consommation totale
idle
ALU
Mult
RAM read
RAM write
Cache read
Cache write
1 : 100 MHz PXbus 50 MHz sans DMA 0.9V
2 : 100 MHz PXbus 50 MHz avec DMA 0.9V
3 : 200 MHz PXbus 50 MHz sans DMA 1.0V
4 : 200 MHz PXbus 50 MHz avec DMA 1.0V
5 : 200 MHz PXbus 100 MHz sans DMA 1.3V
6 : 200 MHz PXbus 100 MHz avec DMA 1.3V
800
600
400
200
1
2
3
4
5
6
test


Comparaison des performances relatives
1 : 100 MHz PXbus 50 MHz sans DMA 0.9V
2 : 100 MHz PXbus 50 MHz avec DMA 0.9V
3 : 200 MHz PXbus 50 MHz sans DMA 1.0V
4 : 200 MHz PXbus 50 MHz avec DMA 1.0V
5 : 200 MHz PXbus 100 MHz sans DMA 1.3V
6 : 200 MHz PXbus 100 MHz avec DMA 1.3V
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
ALU
Mult
RAM read
RAM write
cache read
cache write
Benchmark

Performances / consommation totale



Consommation et performances : Conclusions

- Importance des mémoires caches
- Importance d'optimiser les logiciels
- Adaptation des fréquences aux besoins :
  - manuelle pour une tâche précise
  - automatique par le système d'exploitation ?
- Tests uniquement sur PXA250. Le PXA255 devrait consommer moins.

Travail effectué

- Jelie
- Adaptation des outils GNU
- Réalisation d'un environnement de développement (C, C++, asm)
- Port de Linux sur Armonie
- Pilote Mubus
- Système multimédia autonome
- Ecran LCD sous Linux
- Analyse de performances / consommation

Conclusion

- Support pédagogique
  - outils de développement
  - système d'exploitation
- Système multimédia embarqué
  - conçu
  - réalisé : alimentation, PCB eMediaKit, LCD
  - à faire : monter eMediaKit, écrire les pilotes pour eMediaKit, boitier, test batteries
- Robot cyclope
  - non testé

Perspectives futures

- Suite d'outils JTAG
  - support d'autres cibles pour Jelie
  - coopération avec l'analyseur de Cédric Gaudin
  - interface JTAG rapide avec FPGA (par exemple analyseur logique de Sébastian Gerlach)
- Robot Cyclope
- Système multimédia embarqué
- Gestion d'énergie sous Linux

Apports personnels

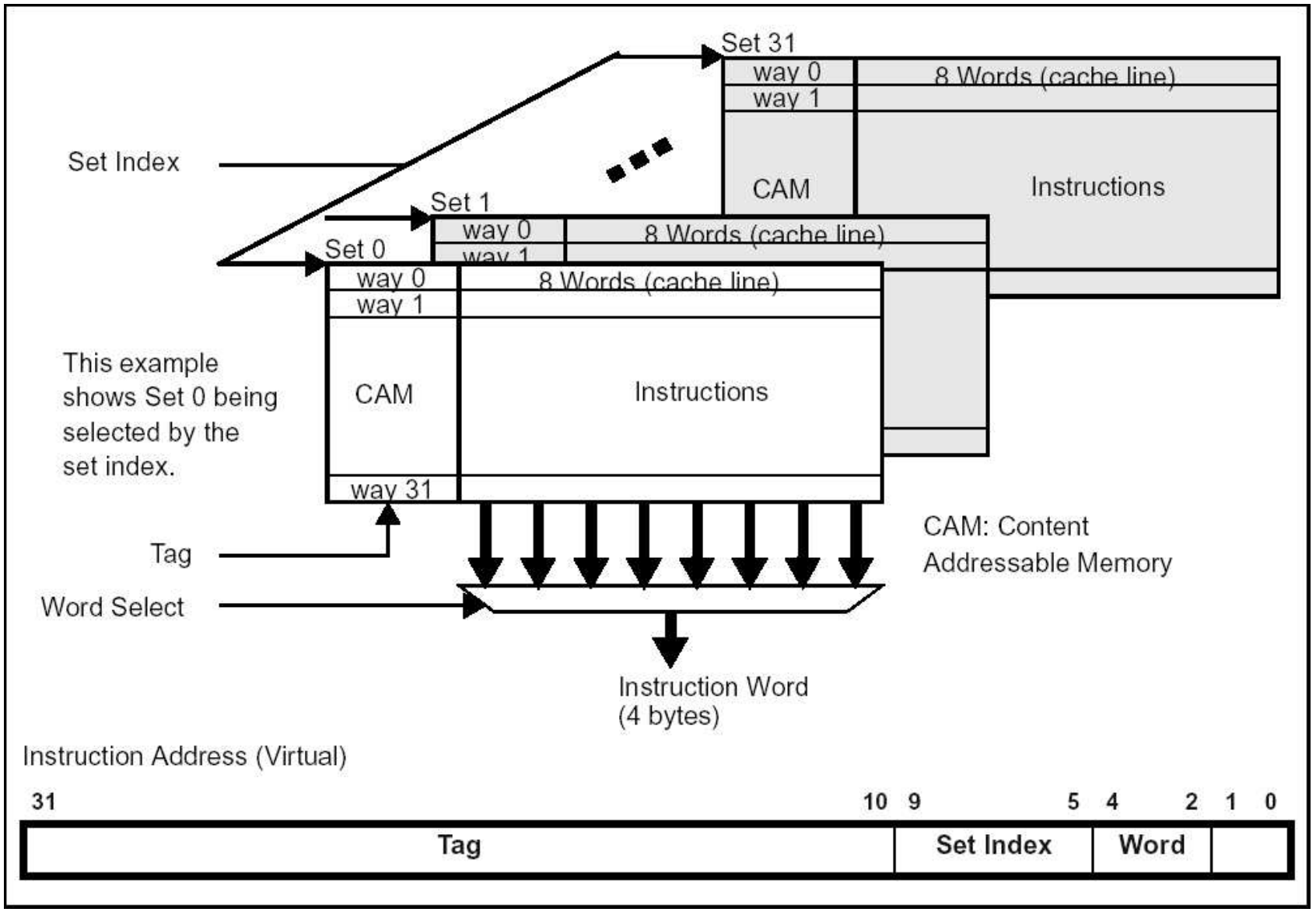
- Excellente ambiance de travail
- Bonne collaboration :
  - dans le groupe,
  - avec les autres projets
- Projet très motivant
- Beaucoup de nouvelles connaissances

Démonstrations

- Développement sans OS
  - débogage avec GDB/Insight
  - test sur afficheur Mubus
- Linux
  - démarrage de Linux depuis la flash
  - test I²C avec PCF8574
  - écran LCD

Avez-vous des questions ?

Organisation de la mémoire cache



Fréquences et alimentations

L	M	Turbo Mode Frequency (MHz) for Values "N" and Core Clock Configuration Register (CCCR[15:0]) programming for Values of "N":				PXbus Frequency	MEM, LCD Frequency (MHz)	SDRAM max Freq
		1.00 (Run)	1.50	2.00	3.00			
27	1	99.5 @.85v	—	199.1 @1.0 V	298.6 @1.1v	50	99.5	99.5
32	1	118.0 @1.0v	—	235.9 @1.1 V	353.9 @1.3v	59	118.0	59.0
36	1	132.7 @1.0v	—	265.4 @1.1v	398.1 @1.3v	66	132.7	66
40	1	147.5 @1.0v	—	294.9 @1.1v	—	74	147.5	74
45	1	165.9 1.0v	—	331.8 1.3v	—	83	165.9	83
27	2	199.1 @1.0v	298.6 @1.1v	398.1 @1.3v	—	99.5	99.5	99.5
32	2	235.9 @1.1v	—	—	—	118	118.0	59.0
36	2	265.4 @1.1v	—	—	—	132.7	132.7	66
40	2	294.9 @1.1v	—	—	—	147.5	147.5	74
45	2	331.9 @1.3v	—	—	—	165.9	165.9	83