

# M1 SESI 2017-2018

## Architecture Multi-Processeurs

### TP1 : Prototypage Virtuel

Kevin Mambu

February 12, 2018

## 1 Énoncé du TP

Le sujet de ce TP sera le prototypage virtuel, de la simulation extrêmement fidèle du système. L'idée est de reproduire le comportement d'un ordinateur au cycle près et au bit près.

## 2 Mode opératoire

On "construit" une machine complète avec son/ses processeur-s en interconnectant entre eux tous les composants matériels d'une vraie machine.

Ces composants existent déjà au sein d'une bibliothèque du domaine public appelée SoClib, utilisant le langage SYSTEM\_C (C++). On aura un processeur de fourni et on développera, dans un premier temps en MIPS32 puis en C, des programmes à exécuter sur notre prototype virtuel.

- Machine simulante : 1GHz (1 milliard inst/s)
- Prototype virtuel : 1Mhz (1 million inst/s)

Le prototype est de plus faible cadence afin de pouvoir rendre l'évolution de la machine plus macroscopique (on peut faire évoluer le système cycle par cycle) et de pouvoir inspecter de manière plus fine.

### 3 Question C1/C2

A	$SEL \bullet ADR\_OK \bullet DELAY \bullet \overline{READ}$
B	$SEL \bullet \overline{READ} \bullet \overline{DELAY} \bullet ADR\_OK$
C	$SEL \bullet READ \bullet \overline{\overline{DELAY}} \bullet ADR\_OK$
D	$SEL \bullet ADR\_OK \bullet DELAY \bullet READ$
E	$SEL \bullet \overline{ADR\_OK}$
F	$\overline{SEL}$
G	
R	$SEL \bullet \overline{ADR\_OK}$
S	$SEL \bullet ADR\_OK$
T	$\overline{SEL} \bullet ADR\_OK$
U	$GO$
V	$GO$
U'	$\overline{GO}$
V'	$\overline{GO}$
X	$SEL \bullet ADR\_OK$
Y	$SEL \bullet \overline{ADR\_OK}$
Z	$\overline{SEL} \bullet ADR\_OK$

	ACK_EN	ACK_VAL	DT_EN	MEM_CMD
<b>IDLE</b>	X	0	NOP	0
<b>R_WAIT</b>	WAIT	1	NOP	0
<b>R_OK</b>	READY	1	READ	1
<b>W_WAIT</b>	WAIT	1	NOP	0
<b>W_OK</b>	READY	1	WRITE	1
<b>ERROR</b>	ERROR	1	NOP	0

## 4 Question D1/D2

Label	Expr.	I	
A		J	$\overline{RDY}$
B	$GNT$	K	$RDY \bullet LAST$
B'	$\overline{GNT}$	L	$RDY \bullet \overline{LAST}$
C		M	$GNT$
D	$RDY$	M'	$\overline{GNT}$
D'	$\overline{RDY}$	N	
E	$RDY$	O	$\overline{RDY}$
E'	$\overline{RDY}$	P	$RDY \bullet NUL$
F	$RDY$	Q	$RDY \bullet \overline{NUL}$
F'	$\overline{RDY}$	R	$GNT$
G	$RDY$	R'	$\overline{GNT}$
G'	$\overline{RDY}$	S	
H	$GNT$	T	$RDY$
H'	$\overline{GNT}$	T'	$\overline{RDY}$

	REQ	CMD_EN	ADR_VALUE	READ_VALUE	LOCK_VALUE	DT_EN
<b>INIT</b>	0	0	X	X	X	0
<b>RAM_REQ</b>	1	0	X	X	X	0
<b>RAM_A0</b>	0	1	RAM_BASE	1	1	0
<b>RAM_A1_D0</b>	0	1	$RAM\_BASE + 4$	1	1	0
<b>RAM_A2_D1</b>	0	1	$RAM\_BASE + 8$	1	1	0
<b>RAM_A3_D2</b>	1	1	$RAM\_BASE + 12$	1	0	0
<b>RAM_D3</b>	0	0	X	1	0	0
<b>W_REQ</b>	1	0	X	X	X	0
<b>W_AD</b>	0	1	$SEG\_TTY\_BASE$	0	0	0
<b>W_DT</b>	0	1	$SEG\_TTY\_BASE$	0	0	1
<b>STS_REQ</b>	1	0	X	X	0	0
<b>STS_AD</b>	0	1	$SEG\_TTY + 4$	X	X	0
<b>STS_DT</b>	0	1	X	?	X	?
<b>BUF_REQ</b>	1	0	X	X	X	0
<b>BUF_AD</b>	0	1	X	X	X	1
<b>BUF_DT</b>	0	1	X	1	X	1

## 5 Question E1/E2

Label	Expr.
X'	$\overline{REQ}$
X	$REQ$
Y'	$\overline{LOCK}$
Y	$LOCK$
Z'	$\overline{(ACK \oplus WAIT)} + LOCK$
Z	$\overline{LOCK} \bullet (ACK \oplus WAIT)$
K	$\overline{REQ}$
L	$REQ$

	GNT	SEL0	SEL1
IDLE	0	0	0
AD	1	$(A \gg msb\_shift) == 0$	$(A \gg msb\_shift) == 1$
DTAD	1	$(A \gg msb\_shift) == 0$	$(A \gg msb\_shift) == 1$
DT	0	0	0

## 6 Question E3

Lorsque l'état courant du BCU est à l'état IDLE, si après envoi de la dernière réponse il restait une requête suivante dans l'ordre de priorité, il faut la traiter au cycle suivant. Le BCU ne retourne dans l'état IDLE que lorsqu'il est en attente d'une requête maître.