SFO Correction de Mich

Pour voir votre correction, cliquer sur le lien ci-dessous (Le fichier RomProgramme.ass a éventuellement été modifié pour supprimer les erreurs d'assemblage, les lignes corrigées débutent par ";$") :

* [interro79](http://intranet-gei.insa-toulouse.fr:8181/Enseignements/SFO/IMACS-3/Documents/ControleSession1/ComptesInterro/interro79)

Pour l'exercice 1 (Désassemblage), voici une solution possible :

;=====================

; Exercice Desassemblage

;=====================

ORG 0x20

deb1: LDI R0, #0x0000; --> WORD 0x4100; WORD 0x0000

LDI R1, #0x0001; --> WORD 0x4110; WORD 0x0001

LDI R2, #0xACC0; --> WORD 0x4120; WORD 0xACC0

etiq2: LSR R2, R2; --> WORD 0x2022

BNC etiq1; --> WORD 0x5E02

ADD R0, R1; --> WORD 0x1101

etiq1: BNZ etiq2; --> WORD 0x57F8

ST R0, [#DIGITHEXA]; --> WORD 0x4200; WORD 0x4004

fin1: B fin1; --> WORD 0x51FE

Pour l'exercice 2 (microprogrammation de l'instruction LDRBX), voici une solution possible :

RomMicrocode :

<Word adr="96" val="0x0\_6800\_003C\_1001\_0C60/ LDRBX BR et sauvegarde PC et Rb dans ALU"/>

<Word val="0x0\_0000\_003F\_C804\_0000/ PC = M[PC]"/>

<Word val="0x1\_6100\_003C\_1800\_0000/ PC = PC + Rb"/>

<Word val="0x0\_0000\_0083\_C80C\_0000/ Ra = (octet) M[PC]"/>

<Word val="0x1\_0A00\_003C\_0800\_0000/ PC = savePC dans ALU"/>

<Word val="0x1\_2100\_023F\_C802\_08F0/ PC = PC+2, go fetch"/>

<Word val="0x0\_0000\_0000\_0000\_0000/ "/>

RomTranscode :

<Word adr="101" val="0x60/LDRBX "/>

Pour l'exercice 3 (IT), voici une solution possible :

;=====================

; Exercice traitement d'IT

;=====================

ORG 0x0050

IT5:  ;; handler IT5

LDI R1, #0x0004 ; msque BT Down

IT51: LDB R0, [#BP] ; lecture bouton

AND R0, R1

BNZ IT51 ; attente bouton relevé

LDI R1, #0x0001 ; R4 = Complement à 2 de R4

NOT R4, R4

ADD R4, R1

RTI

deb3: LDI R14, #PtrIT0;

LDI R0, #0b00100000 ; demasquer IT5

STB R0, [#IMR]

suite3:

Correction Pix Jordan

Mode temps simulé !

Lilas prêt !

Script lancé

/instanceVariable/circuit/ClkGen\_1493103059897/clkFX : 66 Mhz

[30mChargement RomMicrocode[0m

[30mChargement RomTranscode[0m

[30mChargement RomProgramme[0m

[30mSimulateur Correcteur prêt ![0m

-----------------------------------

NEGATU MICHAEL

-----------------------------------

Nbr erreurs assembleur : 0 pénalité=0.0

-----------------------------------

Evaluation Desassembleur

Question Desassembleur : 6.0/6

;

-----------------------------------

Evaluation Microprogrammation

[36mDecode[0m

[36mFetch[0m

R15 mal initialisé, R15=0x0088 attendu 0x008A

Exercice Microprogrammation : 6.0/8

-----------------------------------

Evaluation Desassembleur

Question traitement d'IT : 6.0/6

; mais faudrait sauvegarder R8 dans IT5 avant de l'utiliser !

-----------------------------------

Note globale = 18.0 /20

-----------------------------------

Recapitulatif=NEGATU,MICHAEL,0,0.0,6.0,6.0,6.0,18.0,18.0

-----------------------------------

Lilas terminé !

==========================================================

Votre fichier RomProgramme.ass :

;; ===================================================================

;;

;; Nom : Negatu

;; Prénom : Michael

;;

;; ===================================================================

ORG 0

WORD 0x0400 ;; initialisation SP (premier mot suivant le dernier mot de la ram)

WORD deb1 ; initialisation PC.

PtrIT0: WORD errIT

PtrIT1: WORD errIT

PtrIT2: WORD errIT

PtrIT3: WORD errIT

PtrIT4: WORD errIT

PtrIT5: WORD IT5

PtrIT6: WORD errIT

PtrIT7: WORD errIT

errIT: B errIT ; pour bloquer le programme si une IT imprévue est levée

;-------------------

; Interface OCD

;-------------------

SW: EQU 0x4000 ; Switch carte Nexys3\_IO (octet)

BP: EQU 0x4001 ; Bouton Poussoir carte Nexys3\_IO (octet)

LED: EQU 0x4002 ; Leds carte Nexys3\_IO (octet)

DIGITHEXA:EQU 0x4004 ; Afficheurs 7 segments carte Nexys3\_IO (mot)

;-------------------

; Interface NVIC

;-------------------

ITVR: EQU 0x2000 ; noIT (octet)

IMR: EQU 0x2001 ; masque des IT (octet)

;=====================

; Exercice Desassemblage

;=====================

;$A

ORG 0x20

deb1: LDI R0,#0x0000 ;;WORD 0x4100

;;WORD 0x0000

LDI R1,#0x0001 ;; WORD 0x4110

;;WORD 0x0001

LDI R2,#0xAAC0 ;;WORD 0x4120

;;WORD 0xACC0

loop: LSR R2,R2 ;; WORD 0x2022

BNC suite ;; WORD 0x5E02

ADD R0,R1 ;; WORD 0x1101

suite: BNZ loop ;; WORD 0x57F8

ST R0,[#0x4004] ;; WORD 0x4200

;; WORD 0x4004

fin1: B fin1 ;; WORD 0x51FE

;=====================

; Exercice microProgrammation

;=====================

ORG 0x0040

table: WORD 0x3C50 ; table de 4 octets

WORD 0x6E32

deb2: ;; Test de l'instruction LDRBX

LDI R1, #0x0003

WORD 0x6521 ; codage de LDRBX R2, [#table,R1]

WORD 0x0040 ; à l'aide de 2 directives WORD

fin2: B fin2

;=====================

; Exercice traitement d'IT

;=====================

;$A mais faudrait sauvegarder R8 dans IT5 avant de l'utiliser !

ORG 0x0050

IT5:

NOT R4,R4

LDI R8,#0x0001

ADD R4,R8

RTI ;; handler IT5

deb3: LDI R14,#PtrIT0 ;;

LDI R0,#0x20

STB R0,[#IMR]

suite3:

;;; NE PAS MODIFIER LE CODE D'ICI JUSQU'A LA FIN DU FICHIER

;;; |

;;; \|/

;;; V

;; Programme inrcrémentation/décrémentation du registre R3

LDI R3, #0 ; R3 := 0

LDI R4, #1 ; R3 := R1

fin3: ADD R3, R4

B fin3