

Instructions

- Durée 1 heure.
- Lisez bien toutes les questions.
- **Tout votre code sera déposé dans deux fichiers sur TOMUSS.**
- Tout document interdit, pas de navigateur web, sauf TOMUSS.

Exercice 1 – Circuits combinatoires et séquentiels (travail à réaliser avec LogiSim)

A. Circuits combinatoires

Les circuits sont à réaliser avec des portes NAND seulement (vous avez cependant le droit d'utiliser des portes d'entrances variées).

1. Proposez un circuit réalisant un multiplexeur comportant une entrée de contrôle 1 bit, deux entrées de données 1 bit et une sortie 1 bit.
2. Proposez un circuit réalisant un demi-additionneur, c'est-à-dire un circuit prenant en **entrées** deux bits a_1 et b_1 , et ayant en **sorties** un bit s_1 représentant la somme de a_1 et b_1 ainsi qu'un bit représentant la retenue générée r_1 .
3. Proposez un circuit réalisant, à l'aide des demi-additionneurs que vous avez réalisés à la question précédente, un circuit incrémenteur 4 bits. Ce circuit prend en entrée 5 bits e_3, e_2, e_1, e_0 et inc , et a en sortie 5 bits s_3, s_2, s_1, s_0 et deb . Les nombres X ($e_3e_2e_1e_0$) et Y ($s_3s_2s_1s_0$) sont des entiers non signés compris entre 0 et 15.
 - Si $inc = 0$, alors $Y = X$ et $deb = 0$
 - Si $inc = 1$ et $X \neq 15$, alors $Y = X+1$ et $deb = 0$
 - Si $inc = 1$ et $X = 15$, alors $Y = X$ et $deb = 1$

B. Circuit séquentiel

4. Proposez un circuit réalisant un compteur par 5 qui passe successivement par les valeurs 0 – 2 – 4 – 6 – 8 avec un nombre minimal de bascules. Tester ce circuit à l'aide de l'afficheur Hex Digit.

Vous déposerez votre fichier .circ (avec la partie combinatoire et la partie séquentiel) dans la case TOMUSS TPN_rendu_circ.

Exercice 2 – Programmation en langage d'assemblage (travail à réaliser avec PennSim)

On dispose d'un tableau d'entiers signés représentés en complément à 2 sur 16 bits. Le tableau débute en mémoire à l'adresse désignée par le label `tabdeb` et termine à l'adresse désignée par le label `tabfin`.

Écrivez un programme en langage d'assemblage du LC-3 qui permet de trouver la valeur MIN et la valeur MAX de ce tableau. Les valeurs MIN et MAX seront rangées en mémoire aux labels `min` et `max`.

Récupérez dans TOMUSS le fichier .asm à compléter dans la case TPN_sujet_asm.

Vous déposerez votre fichier .asm complété dans la case TOMUSS TPN_rendu_asm.

Syntaxe	action	NZP	codage															
			opcode				arguments											
			F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
NOT DR, SR	DR <- not SR	*	1	0	0	1		DR		SR		1	1	1	1	1	1	1
ADD DR, SR1, SR2	DR <- SR1 + SR2	*	0	0	0	1		DR		SR1		0	0	0			SR2	
ADD DR, SR1, Imm5	DR <- SR1 + SEXT(Imm5)	*	0	0	0	1		DR		SR1		1				Imm5		
AND DR, SR1, SR2	DR <- SR1 and SR2	*	0	1	0	1		DR		SR1		0	0	0			SR2	
AND DR, SR1, Imm5	DR <- SR1 and SEXT(Imm5)	*	0	1	0	1		DR		SR1		1				Imm5		
LEA DR,label	DR <- PC + SEXT(PCoffset9)	*	1	1	1	0		DR								PCoffset9		
LD DR,label	DR <- mem[PC + SEXT(PCoffset9)]	*	0	0	1	0		DR								PCoffset9		
ST SR,label	mem[PC + SEXT(PCoffset9)] <- SR		0	0	1	1		SR								PCoffset9		
LDR DR,BaseR,Offset6	DR <- mem[BaseR + SEXT(Offset6)]	*	0	1	1	0		DR		BaseR						Offset6		
STR SR,BaseR,Offset6	mem[BaseR + SEXT(Offset6)] <- SR		0	1	1	1		SR		BaseR						Offset6		
BR[n][z][p] label Si (cond)	PC <- PC + SEXT(PCoffset9)		0	0	0	0		n	z	p						PCoffset9		