NOM : PRENOM:

UNIVERSITÉ DE RENNES 1

CC SYS1 **Janvier 2021**

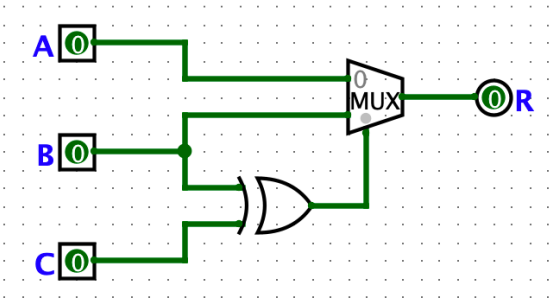
M1 CCN

- Durée de l'épreuve : 2 heures
- Documents de cours, TD et TP autorisés
- Calculatrices non connectées autorisées.
- Le barème est donné à titre purement indicatif.

Les réponses doivent être portées sur ce document, sous peine d'être ignorées par le correcteur.

Exercice 1 : circuits combinatoires (2 points)

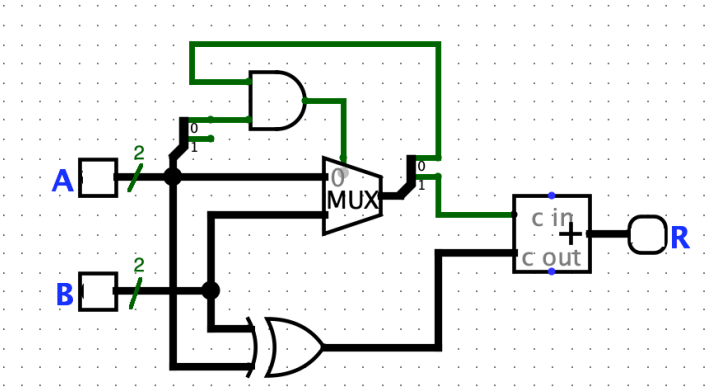
Complétez la table de vérité pour le circuit combinatoire ci-dessous :



A	B	C		R
0	0	0		0
0	0	1		
0	1	0		1
0	1	1		
1	0	0		
1	0	1		0
1	1	0		
1	1	1		1

Exercice 2 : circuits combinatoires (2 points)

Le circuit ci-dessous est-il un circuit combinatoire valide ? Justifiez votre réponse en pointant sur le schéma (et en les expliquant ci-dessous) les problèmes éventuels.



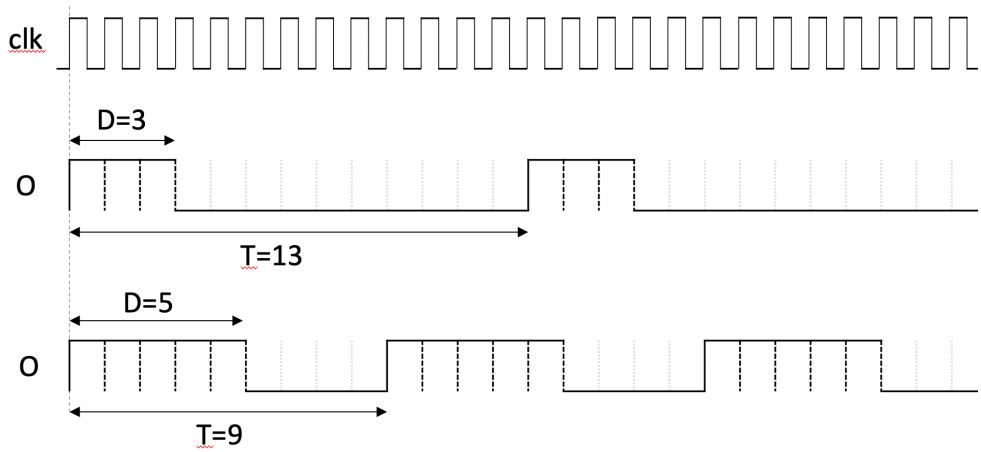
NOM : PRENOM:

Exercice 3 : conception d'un circuit séquentiel (5 points)

On souhaite réaliser un circuit de *Pulse Width Modulation* qui permet de produire un signal de sortie binaire O dont on peut contrôler le nombre moyen de valeurs à "1". Ce composant dispose, en plus du signal clk, de deux entrées T et D.

- Le signal de commande T (codé sur 4 bits) contrôle la période (en nombre de cycles) du signal de sortie
- Le signal de commande D (codé sur 4 bits) contrôle le *Duty Cycle*, c'est-à-dire le nombre de de cycles dans une période ou la sortie est à 1.

Le fonctionnement du composant est illustré sur les chronogrammes ci-dessous, pour différentes valeurs de T et D.



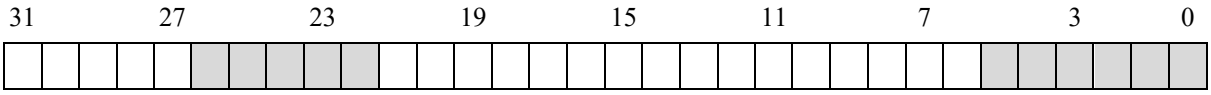
Proposez un circuit séquentiel réalisant cette fonctionnalité.

NOM : PRENOM:

Exercice 4 : codage d'Instructions machine (2,5 points)

On s'intéresse au codage binaire de l'instruction **`add r3,r4,r5`**. Quel est la famille de cette instruction (I, R, J) ? Justifiez votre réponse.

Donnez le codage binaire de l'instruction



Déduisez-en sa représentation en hexadécimal

Exercice 5 : Jeu d'instructions du NIOS2 (4 points)

On s'intéresse à la séquence d'instructions suivante, qui opère à partir d'une mémoire dont le contenu est indiqué ci-dessous.

```
addi r2,zero,0x10
addi r3,r2,0xFFF4
ldb r4,0x4(r3)
ldw r5,0x4(zero)
stw r3,0x0(zero)
slli r6,r2,4
```

	0x0	0x1	0x2	0x3
0x000	0x41	0x42	0x43	0x44
0x004	0x45	0x46	0x47	0x48
0x008	0x80	0x00	0x12	0x76
0x00C	0x03	0x00	0x00	0x00

Donnez les valeurs des registres et le nouveau contenu de la mémoire à l'issu de l'exécution de cette série d'instructions.

r2	
r3	
r4	
r5	
r6	

	0x0	0x1	0x2	0x3
0x000				
0x004				
0x008				
0x00C				

NOM : PRENOM:

Exercice 6 : langage d'assemblage (4 points)

Soit l'extrait d'un source assembleur fourni ci-dessous. Complétez l'image mémoire (une table ASCII est fournie en annexe) en supposant que la zone de données (.data) commence à l'adresse 0, et indiquez les valeurs associées aux identifiants qui y sont définis.

```
.       .data
a       .word -1
b       .byte 0x10
c       .string "ABCDEF"
d       .equ 12
        .align 4
e       .word 0x123456
```

Identifiant	valeur
a	
b	
c	
d	
e	

	0x0	0x1	0x2	0x3
0x000				
0x004				
0x008				
0x00C				
0x010				

Exercice 7 : programmation assembleur (4 points)

On souhaite coder en langage machine une fonction **max2** qui retourne le maximum entre deux valeurs entières passées en paramètres et dont code est fourni ci-après. On supposera que les arguments a et b sont passés par via r4 et r5 et que la valeur de retour est passée dans r2. Proposez une traduction assembleur de cette fonction.

```
int max2(int a, int b) {
      if (a>b) {
            return a;
      } else {
            return b;
      }
}
```

```
max2:
```

Traduisez ensuite en assembleur l'instruction ci-dessous qui appelle la fonction max2 et affiche son résultat sur la sortie standard via un appel système.

```
int main() {
   print_int(max(34,56));
}
```

```
main:
```

NOM : PRENOM:

ANNEXE

Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value
00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h	78	x
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C	|
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL