ALGO
QCM

1. **Un graphe peut être ?**
   (a) Orienté
   (b) Non orienté
   (c) A moitié orienté
   (d) Désorienté

2. **Une collision secondaire représente une collision ?**
   (a) avec coincïdence de valeur de hachage entre un x égal à un y
   (b) sans coincïdence de valeur de hachage entre un x égal à un y
   (c) sans coincïdence de valeur de hachage entre un x différent d'un y
   (d) avec coincïdence de valeur de hachage entre un x différent d'un y

3. **La fonction d'essais successifs est utilisée dans le cas de hachage ?**
   (a) Direct
   (b) Linéaire
   (c) avec Chaînage séparé
   (d) Coalescent

4. **Quelles méthodes de hachage ne sont pas des méthodes indirectes de gestion des collisions ?**
   (a) Hachage linéaire
   (b) double hachage
   (c) Coalescent
   (d) Avec chaînage séparé

5. **Dans un graphe orienté, le sommet x est adjacent au sommet y si ?**
   (a) Il existe un arc (x,y)
   (b) Il existe un arc (y,x)
   (c) Il existe un chemin (x,..,y)
   (d) Il existe un chemin (y,..,x)

6. **Pour les méthodes de hachage, la complexité au pire de la recherche est ?**
   (a) constante
   (b) logarithmique
   (c) linéaire
   (d) quadratique
   (e) exponentielle

1

7. **Quelle méthode de recherche est totalement inadaptée à la recherche par intervalle ?**
   (a) séquentielle
   (b) dichotomique
   (c) ABR
   (d) Arbres équilibrés
   (e) hachage

8. **Quelles méthodes de hachage sont des méthodes indirectes de gestion des collisions ?**
   (a) Hachage linéaire
   (b) double hachage
   (c) Coalescent
   (d) Avec chaînage séparé

9. **Quelle méthode de hachage génère des collisions secondaires ?**
   (a) Hachage linéaire
   (b) double hachage
   (c) Coalescent
   (d) Avec chaînage séparé

10. **L'ordre d'un graphe orienté est ?**
   (a) Le nombre d'arcs du graphe
   (b) Le nombre de sommets du graphe
   (c) Le coût du graphe
   (d) La liste triée des arcs du graphe

2

QCM N°4

lundi 7 novembre 2016

Question 11

Soient E un $\mathbb{R}$-ev et $f \in \mathscr{L}(E)$ quelconque. Alors f injective ssi

a. $\mathrm{Ker}(f) = \mathrm{Im}(f)$

b. $\mathrm{Im}(f) = \{0\}$

c. $\mathrm{Im}(f) = \emptyset$

d. $\mathrm{Ker}(f) = \emptyset$

e. rien de ce qui précède

Question 12

Soient E un $\mathbb{R}$-ev et $(f, g) \in \left(\mathscr{L}(E)\right)^2$ quelconque. Alors

a. $\mathrm{Ker}(g) \subset \mathrm{Ker}(g \circ f)$

b. $\mathrm{Ker}(g \circ f) \subset \mathrm{Ker}(f)$

c. $\mathrm{Im}(f) \subset \mathrm{Im}(g \circ f)$

d. $\mathrm{Im}(g \circ f) \subset \mathrm{Im}(f)$

e. rien de ce qui précède

Question 13

Soient F et G deux sev quelconques d'un $\mathbb{R}$-ev E. On suppose $E = F \oplus G$. Alors

a. $F \cup G = \{0\}$

b. $F \cap G = \emptyset$

c. $F \cup G = \emptyset$

d. rien de ce qui précède

Question 14

Soit $E = \{P \in \mathbb{R}[X],\ d^\circ(P) = 2\}$. Alors E est un $\mathbb{R}$-ev.

a. vrai

b. faux

3

Question 15

Soit E l'ensemble des séries numériques convergentes. Alors E est un $\mathbb{R}$-ev.

a. vrai

b. faux

Question 16

Soit $\sum u_n$ une série à termes positifs et $(S_n) = \left(\sum_{k=1}^{n} u_k\right)$. Alors

a. (S_n) est croissante

b. (S_n) est décroissante

c. (S_n) n'est pas nécessairement monotone

d. $\sum u_n$ converge ssi (S_n) est majorée

e. rien de ce qui précède

Question 17

a. $\sum \frac{(-1)^n}{n}$ converge

b. $\sum \frac{(-1)^n}{n}$ converge absolument

c. $\sum \frac{1}{n}$ converge

d. rien de ce qui précède

Question 18

Soit (u_n) une suite réelle telle que $u_n \underset{+\infty}{\sim} \frac{(-1)^n}{n}$. Alors

a. $\sum u_n$ converge

b. $\sum v_n$ diverge

c. on ne peut rien dire sur la nature de $\sum u_n$

4

Question 19

Soit (u_n) une suite réelle convergente quelconque. Alors

a. $\sum u_n$ converge

b. $\sum(u_n - u_{n-1})$ converge

c. $\sum(u_n - u_{n-1})$ diverge

d. $\sum u_n$ converge absolument

e. rien de ce qui précède

Question 20

Soit (u_n) une suite réelle telle que $\sum u_n$ converge absolument. Alors $\sum u_n$ converge.

a. vrai

b. faux

5

QCM Azar Chap 3 (Pres perf tenses) Choose the appropriate response.

21. There has been snow on the ground ___ Thanksgiving Day.
    a. since
    b. for
    c. during
    d. All of the above.

22. Jean Pierre has studied English ___ less than a year.
    a. since
    b. for
    c. during
    d. None of the above.

23. Choose the correct end for this sentence: I moved to Villejuif...
    a. for two years.
    b. since two years.
    c. last year.
    d. since last year.

24. How long...
    a. have you had that computer?
    b. have you that computer?
    c. have you got that computer?
    d. do you have that computer?

25. Choose the correct end for this sentence: "So far this week....
    a. I've had two tests and a quiz."
    b. I didn't practice guitar."
    c. I am having two tests."
    d. I have not see John."

26. Choose the correct end for this sentence: "I'm really hungry.
    a. I didn't eat since I got up."
    b. I never eat since I got up."
    c. I haven't eaten since I got up."
    d. I haven't ate since I got up."

27. Choose the correct end for this sentence: Last January
    a. I have seen snow for the first time.
    b. I saw snow for the first time.
    c. I have been seeing snow for the first time.
    d. I had seen snow for the first time.

28. "What ____ Dariush ____ for all these hours?"
    a. has / been doing
    b. did / done
    c. is / doing
    d. have / done

29. "I admit that I ____ older ____ I last saw you."
    a. am getting / since
    b. have get / since
    c. have gotten / for
    d. have gotten / since

30. Choose the correct end for this sentence: "Are you taking Advanced Calculus this semester?" "No, I ___it. I ___ last semester."
    a. am already taking / took it
    b. have already taken / had taken it
    c. have already took / had took it
    d. have already taken / took it

6

31. The 'father" of French anthropology is considered to be:
    a. Michel Foucault
    b. Bruno Latour
    c. Claude Lévi-Strauss
    d. Pierre Bourdieu

32. The place Bourdieu did fieldwork:
    a. Morocco
    b. Algeria
    c. Cyprus
    d. United States

33. The model of culture associated with Lévi-Strauss is called:
    a. structuralism
    b. cultural evolution
    c. cultural relativism
    d. fundamentalism

34. Bourdieu's 'habitus' refers to:
    a. the everyday habits and practices that people unconsciously engage in
    b. the environment in which people live
    c. the traditional clothing in a culture
    d. slang speech

35. Michel Foucault's work focuses on understanding structures of _____ in society.
    a. religion
    b. power
    c. education
    d. consumption

36. "Cultural Relativism" refers to the principle that:
    a. how a person acts or believes can be understood in the framework of his/her own culture
    b. all cultures are related
    c. some cultures are primitive relative to others
    d. cultures are difficult to study

37. The 'science' in *science humaine/social science* refers to:
    a. using a rigorous methodology to investigate theories about culture and society
    b. using popular understandings about culture to support scientific theories
    c. studying human beings using the methods to study bacteria in a laboratory
    d. 'science' in this phrase is just a joke

38. The concept of Weltanschauung sees culture and _____ as closely interacting in how a person views the world.
    a. economy
    b. language
    c. music
    d. technology

39. Structuralism was influenced by theories about human _____.
    a. behavior
    b. psychology
    c. genetics
    d. language

40. Modern theories about culture see culture as primarily an aspect of human _____.

a. artistic spirit
b. beauty
c. cognition (thinking)
d. unconscious desires

8

SPE
2016/2017

Q.C.M n°4 de Physique

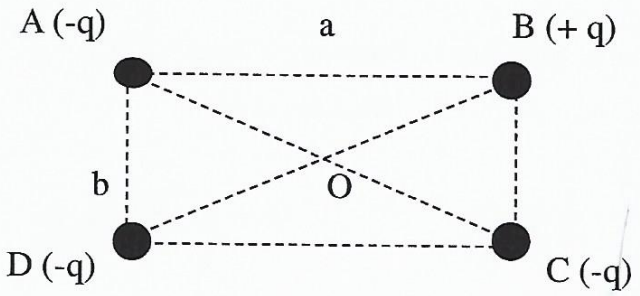
41- Le potentiel électrostatique créé au point M, par une charge q_A placée au point A est donné par :

a) $V_A(M) = k\dfrac{q_A}{(AM)^2}$ b) $V_A(M) = k\dfrac{|q_A|.|q_M|}{(AM)^2}$ c) $V_A(M) = k\dfrac{q_A}{AM}$

42- Un doublet électrique (-Q, +Q) de charges placées respectivement aux points A et B crée un champ électrique au point A de norme :

a) $E(A) = k\dfrac{Q}{(AB)^2}$ b) $E(A) = \dfrac{4kQ}{(AB)^2}$ c) $E(A) = \dfrac{-kQ}{(AB)^2}$ d) $E(A) = 0$

43- On considère la distribution de charges suivante :



Le champ électrique créé au point O : centre du rectangle est

a) orienté vers le point B b) infini c) nul d) orienté vers le point D

44- Dans le schéma ci-dessus, la force électrique exercée sur une charge (-q) que l'on place au centre O du rectangle est

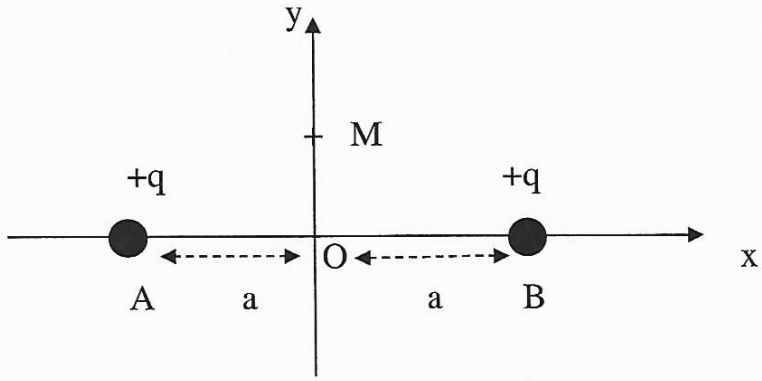
a) nulle b) orientée vers le point B c) orientée vers le point D

45- Dans le schéma de la question (43), le potentiel créé par les quatre charges au point O est

a) $V(O) = 2k\dfrac{q}{OB}$ b) $V(O) = -2k\dfrac{q}{OB}$ c) $V(O) = -k\dfrac{q}{a}$ d) $V(O) = 0$

9

46- On considère la distribution de charges représentée ci-dessous, le champ électrique créé au point O est



a) nul b) orienté vers le point B c) orienté vers le point A

47- Le champ électrique créé au point M, par la distribution de charges représentée ci-dessus est

a) parallèle à l'axe (Ox) b) nul c) porté par l'axe (Oy)

48- La relation champ-potentiel appliquée au potentiel : $V(x,z) = -2x^2 z + \frac{3}{x}$, permet de trouver les composantes du champ électrique données par :

a) $\vec{E} = \begin{pmatrix} 4xz - \frac{3}{x^2} \\ 0 \\ 2x^2 \end{pmatrix}$ b) $\vec{E} = \begin{pmatrix} 4xz + \frac{3}{x^2} \\ 0 \\ 2x^2 \end{pmatrix}$ c) $\vec{E} = \begin{pmatrix} -\frac{3}{x^2} \\ 0 \\ 2x^2 \end{pmatrix}$

49- La circulation du champ électrostatique d'un point A vers un point B est définie par :

a) $C(\vec{E}) = -\vec{grad}(V)$ b) $C(\vec{E}) = \vec{E}.\vec{dl}$ c) $C(\vec{E}) = V(A) - V(B)$

50) Si une distribution de charges sphérique crée au point M un potentiel électrique V(r,θ) alors le champ électrique aura comme composantes :

a) $\vec{E}\begin{pmatrix} 0 \\ 0 \\ E_\varphi \end{pmatrix}$ b) $\vec{E}\begin{pmatrix} E_r \\ 0 \\ E_\varphi \end{pmatrix}$ c) $\vec{E}\begin{pmatrix} 0 \\ E_\theta \\ E_\varphi \end{pmatrix}$ d) $\vec{E}\begin{pmatrix} E_r \\ E_\theta \\ 0 \end{pmatrix}$

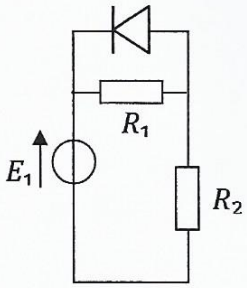
10

QCM Electronique – InfoS3

Pensez à bien lire les questions ET les réponses proposées (attention à la numérotation des réponses)

Q1. Soit le circuit ci-contre, dans lequel on considère la diode idéale :

Choisir l'affirmation correcte si $E_1 = 10V$, $R_1 = 100\Omega$, et $R_2 = 50\Omega$:



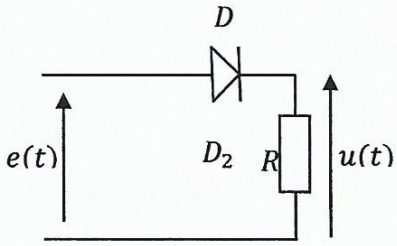
a- La diode est bloquée et la tension à ses bornes est égale à $\left(-\frac{20}{3}\right)$ V.

b- La diode est passante et le courant qui la traverse vaut $100mA$

c- La diode est passante et le courant qui la traverse vaut -5A.

d- La diode est passante et le courant qui la traverse est égal à $200mA$.

Q2. Soit le circuit ci-contre. On considère la diode idéale, et $e(t) = E_0.\sin(\omega.t)$. Choisir l'affirmation correcte :



a- La diode est bloquée et la tension à ses bornes est égale à $\frac{E_0}{R}$ V.

b- Si $e(t) < 0$, alors la diode est passante.

c- Si $e(t) < 0$, alors la diode est bloquée.

d- Si $e(t) > 0$, alors la diode est bloquée.

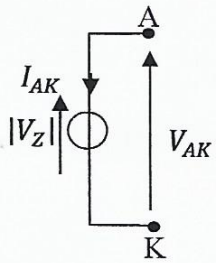
Q3. En polarisation directe, la diode Zéner se comporte comme un générateur de courant.

a- VRAI b- FAUX

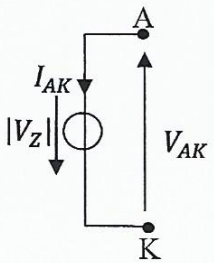
Q4. En polarisation directe, on peut représenter la diode Zéner à l'aide de l'un des 2 modèles : à seuil ou linéaire – le modèle idéal n'existant pas pour cette diode.

a- VRAI b- FAUX

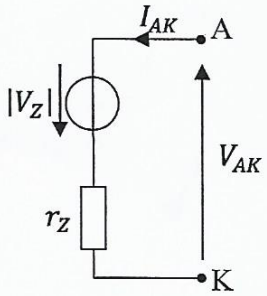
Q5. Par quoi remplace-t-on la diode Zéner lorsqu'elle est passante en inverse si on utilise le modèle à seuil?



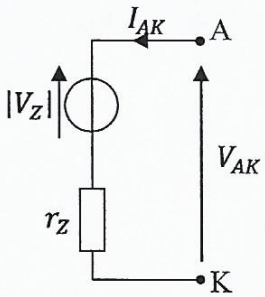
a-



b-



c-



d-

11

Q6. Quel modèle permet la représentation la plus précise de la diode :

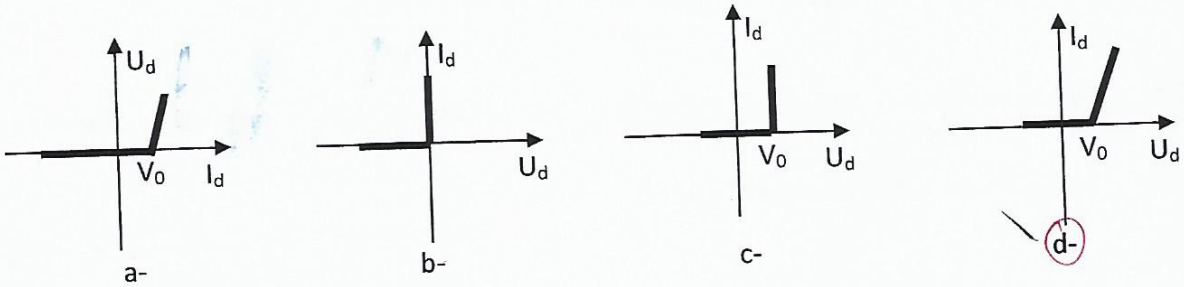
a- Le modèle idéal

b- Le modèle à seuil

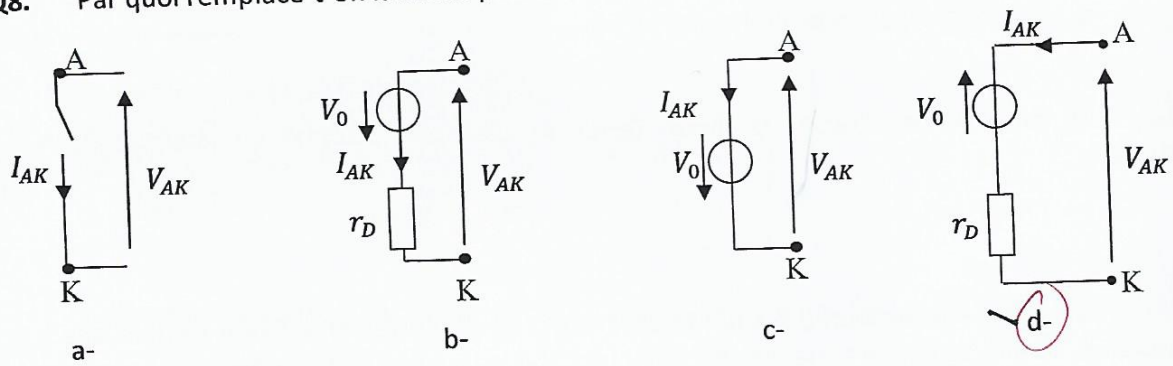
c- Le modèle réel

d- Les trois modèles sont équivalents

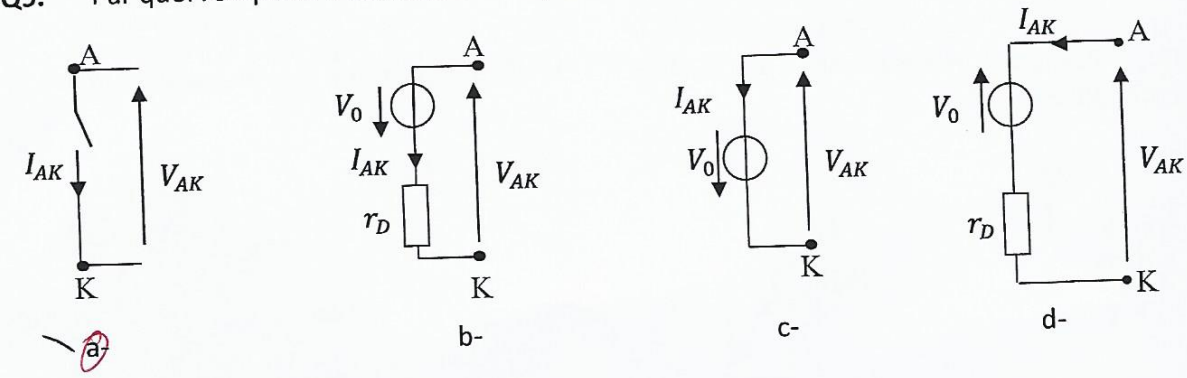
Q7. Laquelle de ces caractéristiques correspond à la caractéristique courant/tension du modèle réel de la diode :



Q8. Par quoi remplace-t-on la diode passante si on utilise le modèle réel?



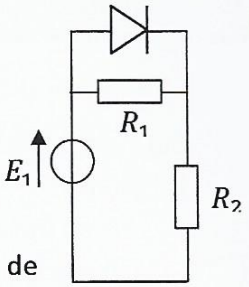
Q9. Par quoi remplace-t-on la diode bloquée si on utilise le modèle à seuil?



12

Q10. Soit le circuit ci-contre, dans lequel on considère la diode est telle que $V_0 = 0{,}6V$:

Choisir l'affirmation correcte si $E_1 = 10V$, $R_1 = 50\Omega$, et $R_2 = 1k\Omega$:



a- La diode est bloquée et la tension à ses bornes est de l'ordre de 0,5V.

b- La diode est passante et le courant qui la traverse est de l'ordre de $10\ mA$

c- La diode est passante et le courant qui la traverse vaut -5A.

d- La diode est passante et le courant qui la traverse est de l'ordre de $9{,}4\ mA$.

13

QCM 4
Architecture des ordinateurs

Lundi 7 novembre 2016

11. Soit l'instruction suivante : `MOVE.W 2(A0),D0`
    A. A0 est incrémenté de 1.
    B. A0 est incrémenté de 4.
    C. A0 ne change pas.
    D. A0 est incrémenté de 2.

12. Quels modes d'adressage ne spécifient pas d'emplacement mémoire ? (deux réponses)
    A. Mode d'adressage immédiat.
    B. Mode d'adressage indirect.
    C. Mode d'adressage direct.
    D. Mode d'adressage absolu.

13. Soient les deux instructions suivantes :

```
TST.B D0
BMI   NEXT
```

    L'instruction `BMI` effectue le branchement si :
    A. D0 = $7F
    B. D0 = $00
    C. D0 = $FF
    D. D0 = $50

14. Soient les deux instructions suivantes :

```
CMP.L D1,D2
BGT   NEXT
```

    L'instruction `BGT` effectue le branchement si :
    A. D2 > D1 (comparaison signée)
    B. D1 > D2 (comparaison signée)
    C. D2 > D1 (comparaison non signée)
    D. D1 > D2 (comparaison non signée)

15. Soient les deux instructions suivantes :

```
CMP.L D1,D2
BLO   NEXT
```

    L'instruction `BLO` effectue le branchement si :
    A. D1 > D2 (comparaison non signée)
    B. D2 > D1 (comparaison signée)
    C. D1 > D2 (comparaison signée)
    D. D2 > D1 (comparaison non signée)

14

16. Si **D0** = $FFFF45BC et **D1**=$FFFF7B44, quelles sont les valeurs des *flags* après l'instruction suivante ? `ADD.B D0,D1`

A. N = 0, Z = 1, V = 1, C = 0
B. N = 0, Z = 1, V = 0, C = 1
C. N = 0, Z = 1, V = 1, C = 1
D. N = 1, Z = 1, V = 0, C = 1

17. Si **D0** = $FFFF45BC et **D1**=$FFFF7B44, quelles sont les valeurs des *flags* après l'instruction suivante ? `ADD.W D0,D1`

A. N = 1, Z = 0, V = 0, C = 1
B. N = 1, Z = 0, V = 1, C = 1
C. N = 0, Z = 0, V = 1, C = 0
D. N = 1, Z = 0, V = 1, C = 0

18. Si **D0** = $FFFF45BC et **D1**=$FFFF7B44, quelles sont les valeurs des *flags* après l'instruction suivante ? `ADD.L D0,D1`

A. N = 1, Z = 0, V = 1, C = 1
B. N = 1, Z = 0, V = 1, C = 0
C. N = 1, Z = 0, V = 0, C = 1
D. N = 0, Z = 0, V = 0, C = 1

19. Soient les cinq instructions suivantes :

```
MOVE.L  (A7)+,D2
MOVE.L  (A7)+,D3
MOVE.L  (A7)+,D4
MOVE.L  (A7)+,A4
MOVE.L  (A7)+,A5
```

Elles sont équivalentes à (une ou plusieurs réponses sont possibles) :

A. MOVEM.L (A7)+,D2-D4/A4/A5
B. MOVEM.L (A7)+,D4/D2/D3/A4/A5
C. MOVEM.L (A7)+,A5/A4-D3/D2/D4
D. MOVEM.L (A7)+,A5/A4/D3/D2/D4

20. Soient les cinq instructions suivantes :

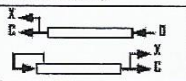
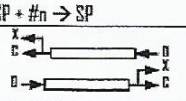
```
MOVE.L  A5,-(A7)
MOVE.L  A4,-(A7)
MOVE.L  D4,-(A7)
MOVE.L  D3,-(A7)
MOVE.L  D2,-(A7)
```

Elles sont équivalentes à (plusieurs réponses possibles) :

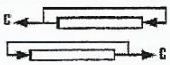
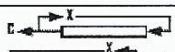
A. MOVEM.L A5/D2-D4/A4,-(A7)
B. MOVEM.L D4/D3/A4/A5,-(A7)
C. MOVEM.L A4/D2-D4,-(A7)
D. MOVEM.L A4-A5/D4/D3/D2,-(A7)

15

EASy68K Quick Reference v1.8 http://www.wowgwep.com/EASy68K.htm Copyright © 2004-2007 By: Chuck Kelly

Opcode	Size	Operand	CCR	Effective Address s=source, d=destination, e=either, i=displacement												Operation	Description
	BWL	s,d	XNZVC	Dn	An	(An)	(An)+	-(An)	(i,An)	(i,An,Rn)	abs.W	abs.L	(i,PC)	(i,PC,Rn)	#n		
ABCD	B	Dy,Dx	*U*U*	e	-	-	-	-	-	-	-	-	-	-	-	$Dy_{10} + Dx_{10} + X \rightarrow Dx_{10}$	Add BCD source and eXtend bit to
		-(Ay),-(Ax)		-	-	-	-	e	-	-	-	-	-	-	-	$-(Ay)_{10} + -(Ax)_{10} + X \rightarrow -(Ax)_{10}$	destination, BCD result
ADD [4]	BWL	s,Dn	*****	e	s	s	s	s	s	s	s	s	s	s	s[4]	s + Dn → Dn	Add binary (ADDI or ADDQ is used when
		Dn,d		e	d[4]	d	d	d	d	d	d	d	-	-	-	Dn + d → d	source is #n. Prevent ADDQ with #n.L)
ADDA [4]	WL	s,An	------	s	e	s	s	s	s	s	s	s	s	s	s	s + An → An	Add address (.W sign-extended to .L)
ADDI [4]	BWL	#n,d	*****	d	-	d	d	d	d	d	d	d	-	-	s	#n + d → d	Add immediate to destination
ADDQ [4]	BWL	#n,d	*****	d	d	d	d	d	d	d	d	d	-	-	s	#n + d → d	Add quick immediate (#n range: 1 to 8)
ADDX	BWL	Dy,Dx	*****	e	-	-	-	-	-	-	-	-	-	-	-	Dy + Dx + X → Dx	Add source and eXtend bit to destination
		-(Ay),-(Ax)		-	-	-	-	e	-	-	-	-	-	-	-	-(Ay) + -(Ax) + X → -(Ax)	
AND [4]	BWL	s,Dn	-**00	e	-	s	s	s	s	s	s	s	s	s	s[4]	s AND Dn → Dn	Logical AND source to destination
		Dn,d		e	-	d	d	d	d	d	d	d	-	-	-	Dn AND d → d	(ANDI is used when source is #n)
ANDI [4]	BWL	#n,d	-**00	d	-	d	d	d	d	d	d	d	-	-	s	#n AND d → d	Logical AND immediate to destination
ANDI [4]	B	#n,CCR	=====	-	-	-	-	-	-	-	-	-	-	-	s	#n AND CCR → CCR	Logical AND immediate to CCR
ANDI [4]	W	#n,SR	=====	-	-	-	-	-	-	-	-	-	-	-	s	#n AND SR → SR	Logical AND immediate to SR (Privileged)
ASL	BWL	Dx,Dy	*****	e	-	-	-	-	-	-	-	-	-	-	-		Arithmetic shift Dy by Dx bits left/right
ASR		#n,Dy		d	-	-	-	-	-	-	-	-	-	-	s		Arithmetic shift Dy #n bits L/R (#n: 1 to 8)
	W	d		-	-	d	d	d	d	d	d	d	-	-	-		Arithmetic shift ds 1 bit left/right (.W only)
Bcc	BW[3]	address[2]	------	-	-	-	-	-	-	-	-	-	-	-	-	if cc true then	Branch conditionally (cc table on back)
																address → PC	(8 or 16-bit ± offset to address)
BCHG	B L	Dn,d	--*--	e[1]	-	d	d	d	d	d	d	d	-	-	-	NOT(bit number of d) → Z	Set Z with state of specified bit in d then
		#n,d		d[1]	-	d	d	d	d	d	d	d	-	-	s	NOT(bit n of d)→ bit n of d	invert the bit in d
BCLR	B L	Dn,d	--*--	e[1]	-	d	d	d	d	d	d	d	-	-	-	NOT(bit number of d) → Z	Set Z with state of specified bit in d then
		#n,d		d[1]	-	d	d	d	d	d	d	d	-	-	s	0 → bit number of d	clear the bit in d
BRA	BW[3]	address[2]	------	-	-	-	-	-	-	-	-	-	-	-	-	address → PC	Branch always (8 or 16-bit ± offset to addr)
BSET	B L	Dn,d	--*--	e[1]	-	d	d	d	d	d	d	d	-	-	-	NOT(bit n of d) → Z	Set Z with state of specified bit in d then
		#n,d		d[1]	-	d	d	d	d	d	d	d	-	-	s	1 → bit n of d	set the bit in d
BSR	BW[3]	address[2]	------	-	-	-	-	-	-	-	-	-	-	-	-	PC → -(SP); address → PC	Branch to subroutine (8 or 16-bit ± offset)
BTST	B L	Dn,d	--*--	e[1]	-	d	d	d	d	d	d	d	d	d	-	NOT(bit Dn of d) → Z	Set Z with state of specified bit in d
		#n,d		d[1]	-	d	d	d	d	d	d	d	d	d	s	NOT(bit #n of d) → Z	Leave the bit in d unchanged
CHK	W	s,Dn	-*UUU	e	-	s	s	s	s	s	s	s	s	s	s	if Dn<0 or Dn>s then TRAP	Compare Dn with 0 and upper bound (s)
CLR	BWL	d	-0100	d	-	d	d	d	d	d	d	d	-	-	-	0 → d	Clear destination to zero
CMP [4]	BWL	s,Dn	-****	e	s[4]	s	s	s	s	s	s	s	s	s	s[4]	set CCR with Dn – s	Compare Dn to source
CMPA [4]	WL	s,An	-****	s	e	s	s	s	s	s	s	s	s	s	s	set CCR with An – s	Compare An to source
CMPI [4]	BWL	#n,d	-****	d	-	d	d	d	d	d	d	d	-	-	s	set CCR with d - #n	Compare destination to #n
CMPM [4]	BWL	(Ay)+,(Ax)+	-****	-	-	-	e	-	-	-	-	-	-	-	-	set CCR with (Ax) - (Ay)	Compare (Ax) to (Ay); Increment Ax and Ay
DBcc	W	Dn,addres[2]	------	-	-	-	-	-	-	-	-	-	-	-	-	if cc false then { Dn-1 → Dn	Test condition, decrement and branch
																if Dn <> -1 then addr →PC }	(16-bit ± offset to address)
DIVS	W	s,Dn	-***0	e	-	s	s	s	s	s	s	s	s	s	s	±32bit Dn / ±16bit s → ±Dn	Dn= [16-bit remainder, 16-bit quotient]
DIVU	W	s,Dn	-***0	e	-	s	s	s	s	s	s	s	s	s	s	32bit Dn / 16bit s → Dn	Dn= [16-bit remainder, 16-bit quotient]
EOR [4]	BWL	Dn,d	-**00	e	-	d	d	d	d	d	d	d	-	-	s[4]	Dn XOR d → d	Logical exclusive OR Dn to destination
EORI [4]	BWL	#n,d	-**00	d	-	d	d	d	d	d	d	d	-	-	s	#n XOR d → d	Logical exclusive OR #n to destination
EORI [4]	B	#n,CCR	=====	-	-	-	-	-	-	-	-	-	-	-	s	#n XOR CCR → CCR	Logical exclusive OR #n to CCR
EORI [4]	W	#n,SR	=====	-	-	-	-	-	-	-	-	-	-	-	s	#n XOR SR → SR	Logical exclusive OR #n to SR (Privileged)
EXG	L	Rx,Ry	------	e	e	-	-	-	-	-	-	-	-	-	-	register ← → register	Exchange registers (32-bit only)
EXT	WL	Dn	-**00	d	-	-	-	-	-	-	-	-	-	-	-	Dn.B → Dn.W | Dn.W → Dn.L	Sign extend (change .B to .W or .W to .L)
ILLEGAL			------	-	-	-	-	-	-	-	-	-	-	-	-	PC→-(SSP); SR→-(SSP)	Generate Illegal Instruction exception
JMP		d	------	-	-	d	-	-	d	d	d	d	d	d	-	↑d → PC	Jump to effective address of destination
JSR		d	------	-	-	d	-	-	d	d	d	d	d	d	-	PC → -(SP); ↑d → PC	push PC, jump to subroutine at address d
LEA	L	s,An	------	-	e	s	-	-	s	s	s	s	s	s	-	↑s → An	Load effective address of s to An
LINK		An,#n	------	-	-	-	-	-	-	-	-	-	-	-	-	An → -(SP); SP → An;	Create local workspace on stack
																SP + #n → SP	(negative n to allocate space)
LSL	BWL	Dx,Dy	***0*	e	-	-	-	-	-	-	-	-	-	-	-		Logical shift Dy, Dx bits left/right
LSR		#n,Dy		d	-	-	-	-	-	-	-	-	-	-	s		Logical shift Dy, #n bits L/R (#n: 1 to 8)
	W	d		-	-	d	d	d	d	d	d	d	-	-	-		Logical shift d 1 bit left/right (.W only)
MOVE [4]	BWL	s,d	-**00	e	s[4]	e	e	e	e	e	e	e	s	s	s[4]	s → d	Move data from source to destination
MOVE	W	s,CCR	=====	s	-	s	s	s	s	s	s	s	s	s	s	s → CCR	Move source to Condition Code Register
MOVE	W	s,SR	=====	s	-	s	s	s	s	s	s	s	s	s	s	s → SR	Move source to Status Register (Privileged)
MOVE	W	SR,d	------	d	-	d	d	d	d	d	d	d	-	-	-	SR → d	Move Status Register to destination
MOVE	L	USP,An	------	-	d	-	-	-	-	-	-	-	-	-	-	USP → An	Move User Stack Pointer to An (Privileged)
		An,USP		-	s	-	-	-	-	-	-	-	-	-	-	An → USP	Move An to User Stack Pointer (Privileged)
	BWL	s,d	XNZVC	Dn	An	(An)	(An)+	-(An)	(i,An)	(i,An,Rn)	abs.W	abs.L	(i,PC)	(i,PC,Rn)	#n		

16

Opcode	Size	Operand	CCR	Effective Address s=source, d=destination, e=either, i=displacement												Operation	Description
	BWL	s,d	XNZVC	Dn	An	(An)	(An)+	-(An)	(i,An)	(i,An,Rn)	abs.W	abs.L	(i,PC)	(i,PC,Rn)	#n		
MOVEA[4]	WL	s,An	-----	s	e	s	s	s	s	s	s	s	s	s	s	s → An	Move source to An (MOVE s,An use MOVEA)
MOVEM[4]	WL	Rn-Rn,d	-----	-	-	d	-	d	d	d	d	d	-	-	-	Registers → d	Move specified registers to/from memory
		s,Rn-Rn		-	-	s	s	-	s	s	s	s	s	s	-	s → Registers	(.W source is sign-extended to .L for Rn)
MOVEP	WL	Dn,(i,An)	-----	s	-	-	-	-	d	-	-	-	-	-	-	Dn → (i,An)...(i+2,An)...(i+4,A.	Move Dn to/from alternate memory bytes
		(i,An),Dn		d	-	-	-	-	s	-	-	-	-	-	-	(i,An) → Dn...(i+2,An)...(i+4,A.	(Access only even or odd addresses)
MOVEQ[4]	L	#n,Dn	-**00	d	-	-	-	-	-	-	-	-	-	-	s	#n → Dn	Move sign extended 8-bit #n to Dn
MULS	W	s,Dn	-**00	e	-	s	s	s	s	s	s	s	s	s	s	±16bit s * ±16bit Dn → ±Dn	Multiply signed 16-bit; result: signed 32-bit
MULU	W	s,Dn	-**00	e	-	s	s	s	s	s	s	s	s	s	s	16bit s * 16bit Dn → Dn	Multiply unsig'd 16-bit; result: unsig'd 32-bit
NBCD	B	d	*U*U*	d	-	d	d	d	d	d	d	d	-	-	-	$0 - d_{10} - X$ → d	Negate BCD with eXtend, BCD result
NEG	BWL	d	*****	d	-	d	d	d	d	d	d	d	-	-	-	0 - d → d	Negate destination (2's complement)
NEGX	BWL	d	*****	d	-	d	d	d	d	d	d	d	-	-	-	0 - d - X → d	Negate destination with eXtend
NOP			-----	-	-	-	-	-	-	-	-	-	-	-	-	None	No operation occurs
NOT	BWL	d	-**00	d	-	d	d	d	d	d	d	d	-	-	-	NOT(d) → d	Logical NOT destination (1's complement)
OR[4]	BWL	s,Dn	-**00	e	-	s	s	s	s	s	s	s	s	s	s[4]	s OR Dn → Dn	Logical OR
		Dn,d		e	-	d	d	d	d	d	d	d	-	-	-	Dn OR d → d	(ORI is used when source is #n)
ORI[4]	BWL	#n,d	-**00	d	-	d	d	d	d	d	d	d	-	-	s	#n OR d → d	Logical OR #n to destination
ORI[4]	B	#n,CCR	=====	-	-	-	-	-	-	-	-	-	-	-	s	#n OR CCR → CCR	Logical OR #n to CCR
ORI[4]	W	#n,SR	=====	-	-	-	-	-	-	-	-	-	-	-	s	#n OR SR → SR	Logical OR #n to SR (Privileged)
PEA	L	s	-----	-	-	s	-	-	s	s	s	s	s	s	-	↑s → -(SP)	Push effective address of s onto stack
RESET			-----	-	-	-	-	-	-	-	-	-	-	-	-	Assert RESET Line	Issue a hardware RESET (Privileged)
ROL	BWL	Dx,Dy	-**0*	e	-	-	-	-	-	-	-	-	-	-	-		Rotate Dy, Dx bits left/right (without X)
ROR		#n,Dy		d	-	-	-	-	-	-	-	-	-	-	s		Rotate Dy, #n bits left/right (#n: 1 to 8)
	W	d		-	-	d	d	d	d	d	d	d	-	-	-		Rotate d 1-bit left/right (.W only)
ROXL	BWL	Dx,Dy	***0*	e	-	-	-	-	-	-	-	-	-	-	-		Rotate Dy, Dx bits L/R, X used then updated
ROXR		#n,Dy		d	-	-	-	-	-	-	-	-	-	-	s		Rotate Dy, #n bits left/right (#n: 1 to 8)
	W	d		-	-	d	d	d	d	d	d	d	-	-	-		Rotate destination 1-bit left/right (.W only)
RTE			=====	-	-	-	-	-	-	-	-	-	-	-	-	(SP)+ → SR; (SP)+ → PC	Return from exception (Privileged)
RTR			=====	-	-	-	-	-	-	-	-	-	-	-	-	(SP)+ → CCR, (SP)+ → PC	Return from subroutine and restore CCR
RTS			-----	-	-	-	-	-	-	-	-	-	-	-	-	(SP)+ → PC	Return from subroutine
SBCD	B	Dy,Dx	*U*U*	e	-	-	-	-	-	-	-	-	-	-	-	$Dx_{10} - Dy_{10} - X \rightarrow Dx_{10}$	Subtract BCD source and eXtend bit from
		-(Ay),-(Ax)		-	-	-	-	e	-	-	-	-	-	-	-	$-(Ax)_{10} - -(Ay)_{10} - X \rightarrow -(Ax)_{10}$	destination, BCD result
Scc	B	d	-----	d	-	d	d	d	d	d	d	d	-	-	-	If cc is true then 1's → d else 0's → d	If cc true then d.B = 11111111 else d.B = 00000000
STOP		#n	=====	-	-	-	-	-	-	-	-	-	-	-	s	#n → SR; STOP	Move #n to SR, stop processor (Privileged)
SUB[4]	BWL	s,Dn	*****	e	s	s	s	s	s	s	s	s	s	s	s[4]	Dn - s → Dn	Subtract binary (SUBI or SUBQ used when
		Dn,d		e	d[4]	d	d	d	d	d	d	d	-	-	-	d - Dn → d	source is #n. Prevent SUBQ with #n.L)
SUBA[4]	WL	s,An	-----	s	e	s	s	s	s	s	s	s	s	s	s	An - s → An	Subtract address (.W sign-extended to .L)
SUBI[4]	BWL	#n,d	*****	d	-	d	d	d	d	d	d	d	-	-	s	d - #n → d	Subtract immediate from destination
SUBQ[4]	BWL	#n,d	*****	d	d	d	d	d	d	d	d	d	-	-	s	d - #n → d	Subtract quick immediate (#n range: 1 to 8)
SUBX	BWL	Dy,Dx	*****	e	-	-	-	-	-	-	-	-	-	-	-	Dx - Dy - X → Dx	Subtract source and eXtend bit from
		-(Ay),-(Ax)		-	-	-	-	e	-	-	-	-	-	-	-	-(Ax) - -(Ay) – X → -(Ax)	destination
SWAP	W	Dn	-**00	d	-	-	-	-	-	-	-	-	-	-	-	bits[31:16]↔bits[15:0]	Exchange the 16-bit halves of Dn
TAS	B	d	-**00	d	-	d	d	d	d	d	d	d	-	-	-	test d→CCR; 1→bit7 of d	N and Z set to reflect d, bit7 of d set to 1
TRAP		#n	-----	-	-	-	-	-	-	-	-	-	-	-	s	PC→-(SSP);SR→-(SSP); (vector table entry) → PC	Push PC and SR, PC set by vector table #n (#n range: 0 to 15)
TRAPV			-----	-	-	-	-	-	-	-	-	-	-	-	-	If V then TRAP #7	If overflow, execute an Overflow TRAP
TST	BWL	d	-**00	d	-	d	d	d	d	d	d	d	-	-	-	test d → CCR	N and Z set to reflect destination
UNLK		An	-----	-	d	-	-	-	-	-	-	-	-	-	-	An → SP; (SP)+ → An	Remove local workspace from stack
	BWL	s,d	XNZVC	Dn	An	(An)	(An)+	-(An)	(i,An)	(i,An,Rn)	abs.W	abs.L	(i,PC)	(i,PC,Rn)	#n		

Condition Tests (+ OR, ! NOT, ⊕ XOR; " Unsigned, ' Alternate cc)					
cc	Condition	Test	cc	Condition	Test
T	true	1	VC	overflow clear	!V
F	false	0	VS	overflow set	V
HI"	higher than	!(C + Z)	PL	plus	!N
LS"	lower or same	C + Z	MI	minus	N
HS", CC'	higher or same	!C	GE	greater or equal	!(N ⊕ V)
LO", CS'	lower than	C	LT	less than	(N ⊕ V)
NE	not equal	!Z	GT	greater than	![(N ⊕ V) + Z]
EQ	equal	Z	LE	less or equal	(N ⊕ V) + Z

An Address register (16/32-bit, n=0-7)
Dn Data register (8/16/32-bit, n=0-7)
Rn any data or address register
s Source, **d** Destination
e Either source or destination
#n Immediate data, **i** Displacement
BCD Binary Coded Decimal
↑ Effective address
1 Long only; all others are byte only
2 Assembler calculates offset
3 Branch sizes: .B or .S -128 to +127 bytes, .W or .L -32768 to +32767 bytes
4 Assembler automatically uses A, I, Q or M form if possible. Use #n.L to prevent Quick optimization

SSP Supervisor Stack Pointer (32-bit)
USP User Stack Pointer (32-bit)
SP Active Stack Pointer (same as A7)
PC Program Counter (24-bit)

SR Status Register (16-bit)
CCR Condition Code Register (lower 8-bits of SR)
N negative, **Z** zero, **V** overflow, **C** carry, **X** extend
* set according to operation's result, = set directly
\- not affected, **0** cleared, **1** set, **U** undefined

Revised by Peter Csaszar, Lawrence Tech University – 2004-2006

Distributed under the GNU general public use license.

17