ALGO
QCM

1. **La fonction d'essais successifs n'est pas utilisée par ?**
   (a) les méthodes indirectes de gestion des collisions
   (b) le hachage avec Chaînage séparé
   (c) le hachage coalescent

2. **La méthode de hachage qui tronçonne la séquence de bits en sous-mots est ?**
   (a) la complétion
   (b) la compression
   (c) l'extraction
   (d) la multiplication

3. **Une fonction de hachage doit être déterministe ?**
   (a) Non
   (b) Oui
   (c) Cela dépend

4. **Le handicap majeur de l'extraction est ?**
   (a) de hacher les anagrammes d'une clé de la même façon
   (b) de nécessiter un m premier majorant le nombre de clés
   (c) de n'utiliser q'une partie de représentation de la clé
   (d) de n'être efficace que sur une petite collection de données

5. **Parmi les méthodes suivantes, lesquelles sont des méthodes de hachage de base ?**
   (a) division
   (b) extraction
   (c) compression
   (d) multiplication

6. **l'efficacité de la multiplication dépend ?**
   (a) principalement de m
   (b) principalement de θ
   (c) autant de m que de θ
   (d) ni de m ni de θ

1

7. **Quelles méthodes sont des méthodes indirectes de gestion des collisions ?**
   (a) le hachage linéaire
   (b) le double hachage
   (c) le hachage coalescent
   (d) le hachage avec chaînage séparé

8. **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

9. **Le double hachage peut générer des collisions secondaires ?**
   (a) Oui
   (b) Non
   (c) quelquefois

10. **Quelles méthodes de hachage utilisent tous les bits de la représentation de la clé ?**
    (a) la complétion
    (b) la compression
    (c) l'extraction
    (d) la division

2

QCM N°3

lundi 10 octobre 2016

Question 11

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 12

Soit (u_n) une suite réelle positive, décroissante et convergeant vers 0. Alors

a. $\sum(-1)^n u_n$ converge

b. $\sum(-1)^n u_n$ diverge

c. on ne peut rien dire sur la nature de $\sum(-1)^n u_n$

Question 13

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 14

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

a. vrai

b. faux

3

Question 15

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

a. $\sum u_n$ converge

b. $\sum u_n$ diverge

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

Question 16

Soit (u_n) une suite réelle quelconque telle que $\sum(u_n - u_{n-1})$ diverge. Alors

a. $\sum u_n$ diverge

b. (u_n) diverge

c. (u_n) converge

d. rien de ce qui précède

Question 17

Soient (u_n) et (v_n) deux suites réelles positives quelconques telles que $u_n = o(v_n)$ et $\sum v_n$ diverge. Alors

a. $\sum u_n$ converge

b. $v_n \longrightarrow +\infty$

c. $\sum u_n$ diverge

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

Question 18

Soit (u_n) une suite réelle strictement positive telle que

$$\frac{u_{n+1}}{u_n} \longrightarrow \frac{1}{4}$$

Alors

a. $\sum u_n$ converge

b. $\sum u_n$ diverge

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

4

Question 19

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 20

Soit $\alpha \in \mathbb{R}$. Alors $\sum n^\alpha$

a. converge ssi $\alpha > 1$

b. converge ssi $\alpha < 1$

c. converge ssi $\alpha < -1$

d. converge ssi $\alpha > -1$

e. diverge pour tout α

5

QCM Azar Chap 2 (Past pres simp/ prog tenses WB prac1-6) Choose the appropriate response.

21. The sun ___ in the west every evening.
    a. set
    b. should set
    c. sets
    d. will set

22. What did you do last night?
    a. I have eaten dinner.
    b. I wrote some email and checked Facebook.
    c. I was watching TV.
    d. I had gone out with some friends.

23. Because of the force of gravity, objects...
    a. fall down and not up.
    b. are falling down.
    c. falls down.
    d. are going to fall down.

A teacher is describing his students' actions from the front of the class (nos. 24 – 30). Choose the logical answer in each case.

24. "Yoko...
    a. writes in her book."
    b. has written in her book."
    c. is writing in her book."
    d. write in her book."

25. "Bill....
    a. scratch his head."
    b. is scratching his head."
    c. will scratch his head."
    d. Scratches his head."

26. "Dariush is staring out the window. He
    a. seem daydreaming...
    b. seem to be daydreaming...
    c. is seeming to be daydreaming...
    d. seems to be daydreaming...

27. "... but perhaps he
    a. thinks hard about verb tenses."
    b. is thinking hard about verb tenses."
    c. has thought hard about verb tenses."
    d. was thinking hard about verb tenses."

28. "What ____ Dariush ____ ?"
    a. you think / is doing
    b. do you think / does
    c. are you thinking / is doing
    d. do you think / is doing

29. "Right now I ____ Nicole."
    a. I look
    b. I look at
    c. I'm looking at
    d. I look to

30. "Nicole ___ angry."
    a. look
    b. looks
    c. is looking
    d. looks like

31. Tylor's definition of culture includes:
    a. arts
    b. morals
    c. religious belief
    d. all of the above

32. 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

33. The person responsible for bringing German philosophical perspectives to American anthropological theories on culture was:
    a. Wilhelm von Humboldt
    b. Franz Boas
    c. Hegel
    d. Edward B. Tylor

34. The article "Shakespeare in the Bush" retells the story of which play:
    a. The Merchant of Venice
    b. Macbeth
    c. Julius Caesar
    d. Hamlet

35. The Tiv are an ethnic group living in:
    a. West Africa
    b. North America
    c. South America
    d. East Asia

36. Bohannon hopes that her telling Shakespeare's story to the Tiv will prove that it has:
    a. universal meaning
    b. different meanings depending on the listener
    c. different meanings depending on the language used
    d. none of the above

37. What is an aspect of the story to which the Tiv react exactly as Bohannon expects?
    a. the appearance of the dead father's ghost
    b. the fact that the chief's brother married his widow
    c. both a and b
    d. none of the above

38. The Tiv interpret the story according to:
    a. How Shakespeare intended it to be understood
    b. universal values about family relationships and political power
    c. their own cultural values and social organization
    d. all of the above

39. "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

40. A cultural relativist would argue that all cultures are inherently _____, even if they differ from one another.
    a. the same
    b. unable to be understood from the outside
    c. equal
    d. boring

SPE
2016/2017

Q.C.M n°3 de Physique

41- La force électrique qui décrit l'interaction entre deux charges ponctuelles q_1 et q_2 , séparées par une distance r est

a) proportionnelle au produit des masses m_1 et m_2 des deux charges.
b) prépondérante à l'échelle atomique
c) inversement proportionnelle au produit des charges
d) inversement proportionnelle à r

42- Un champ électrostatique $\vec{E}$ est dit convergent lorsqu'il est créé par :

a) Un proton
b) Un neutron
c) Un électron

43- Le champ électrostatique $\vec{E}$ créé au point M par une charge placée au même point M est :

a) convergent
b) Nul
c) divergent
d) Non défini

44- Le champ électrostatique $\vec{E}$ créé à l'infini par une charge placée au point O est :

a) convergent
b) Nul
c) Non défini

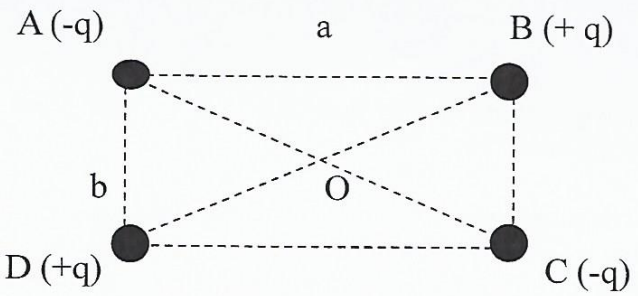
45- L'intensité du champ électrostatique créé au point M, par une charge q_A placée au point A est donné par :

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

46- Un doublet électrique (-Q, +Q) de charges placées respectivement aux points A et B crée un champ électrique au milieu O du segment AB de norme :

a) $E(O) = k\dfrac{Q}{(AB)^2}$ b) $E(O) = \dfrac{4kQ}{(AB)^2}$ c) $E(O) = \dfrac{8kQ}{(AB)^2}$ d) $E(O) = 0$

47- On considère la distribution de charges suivante :



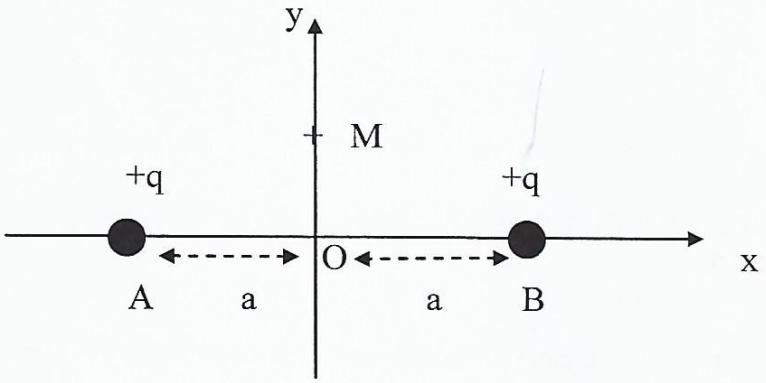
Le champ électrique créé au point O : centre du rectangle est

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

48- 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

49- On considère la distribution de charges suivante :



Le champ électrique créé au point O est

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

50- Le champ électrique créé au point M, par la distribution de charges représentée ci-dessus (question 49) est

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

QCM Electronique – InfoS3

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

Révisions : Lois et Théorèmes de l'électronique

Q51. Soit le circuit ci-contre:

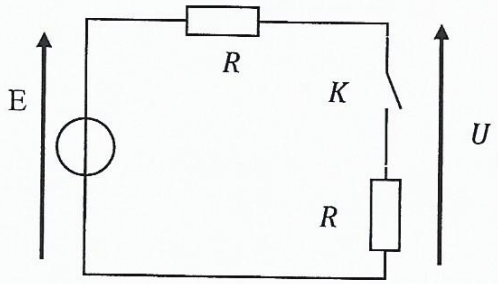
Quelle est la valeur de la tension U lorsque l'intérrupteur K est ouvert?

a- $U = 0$

b- $U = \frac{E}{2}$

c- $U = E$

d- $U = -E$



Les semi-conducteurs et les diodes

Q52. Le dopage permet d'augmenter la conductivité du semi-conducteur

a- VRAI

b- FAUX

Q53. Un semiconducteur intrinsèque est

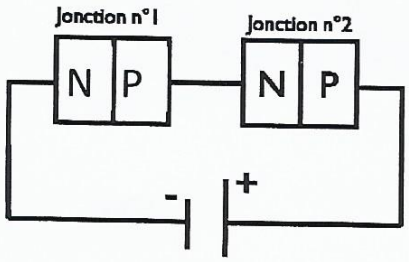
a- Un cristal pur.

b- Un cristal dopé avec des atomes pentavalents

c- Un cristal dopé avec des atomes trivalents

d- Un cristal désordonné.

Q54.



E>2xTension de seuil

Ce circuit est :

a- Passant

b- Bloqué

Q55. L'équation de la caractéristique de la diode s'écrit : $I_D = I_S(e^{\frac{V_D}{mV_T}} - 1)$ où I_D représente le courant qui traverse la diode et V_D, la tension à ses bornes, courant et tension étant fléchés selon la convention récepteur. I_S correspond au courant inverse. C'est un courant :

a- Très grand (plusieurs dizaine 'ampères)

b- Très faible (quelques nano ampères)

Q56. Quel modèle permet la représentation la moins précise de la diode :

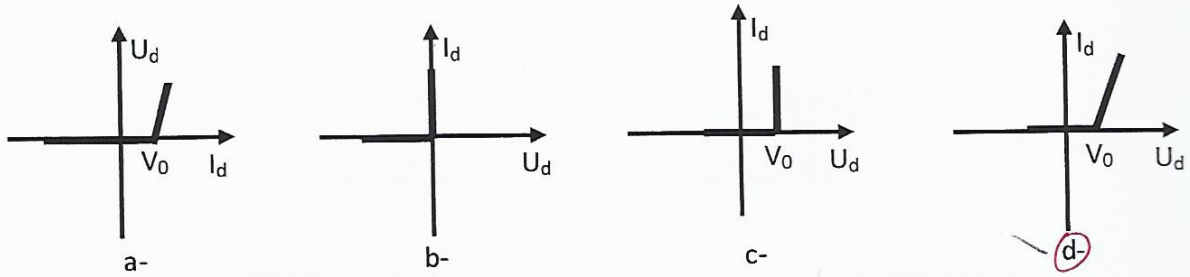
a- Le modèle idéal

b- Le modèle à seuil

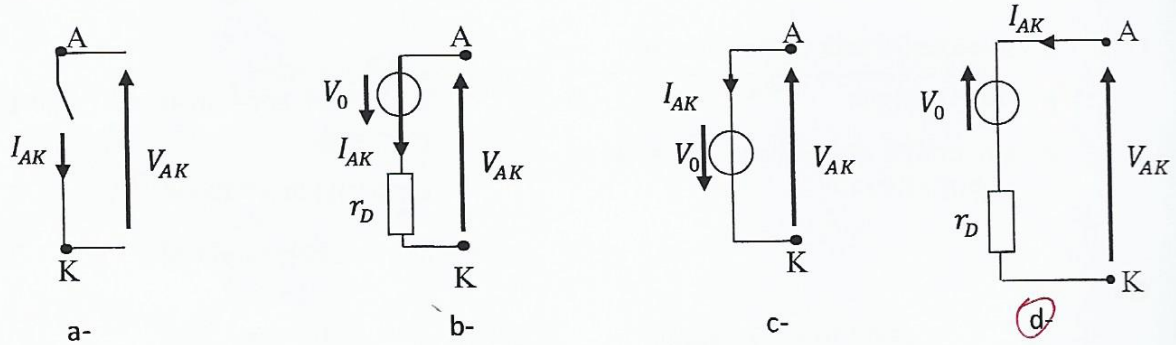
c- Le modèle réel

d- Les trois modèles sont équivalents

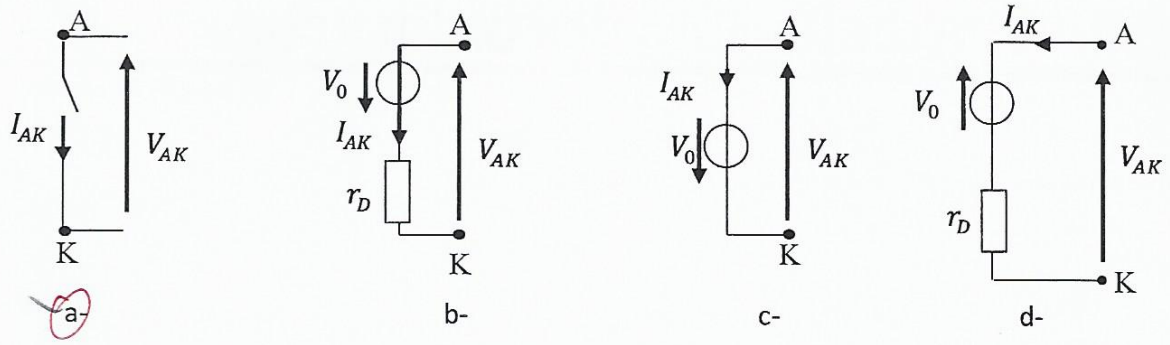
Q57. Laquelle de ces caractéristiques correspond à la caractéristique courant/tension du modèle réel de la diode :



Q58. Par quoi remplace-t-on la diode passante si on utilise le modèle réel?

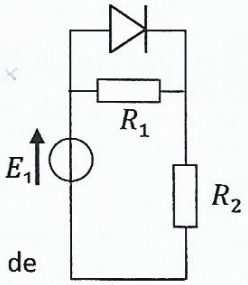


Q59. Par quoi remplace-t-on la diode bloquée si on utilise le modèle à seuil?



Q60. 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 $9{,}4\ mA$.

QCM 3
Architecture des ordinateurs

Lundi 10 octobre 2016

61. Quel mnémonique est une directive d'assemblage ?
    A. ORG
    B. MOVE
    C. ILLEGAL
    D. ADD

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

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

64. Le registre CCR est : (deux réponses)
    A. Sur 8 bits.
    B. Sur 16 bits.
    C. Les 8 bits de poids fort du registre SR.
    D. Les 8 bits de poids faible du registre SR.

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

66. L'instruction `BMI` effectue un branchement si :
    A. N = 0
    B. Z = 1
    C. N = 1
    D. Z = 0

14

67. L'instruction `BNE` effectue un branchement si :
    A. N = 0
    B. Z = 1
    C. N = 1
    D. Z = 0

68. Soient les deux instructions suivantes :

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

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

69. Soient les deux instructions suivantes :

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

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

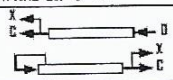
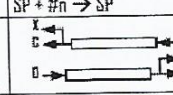
70. Soient les deux instructions suivantes :

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

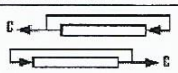
    L'instruction `BLO` effectue le branchement si :
    A. D2 > D1 (comparaison signée)
    B. D1 > D2 (comparaison non signée)
    C. D1 > D2 (comparaison signée)
    D. D2 > D1 (comparaison non signée)

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 -(Ay),-(Ax)	*U*U*	e -	- -	- -	- -	- e	- -	- -	- -	- -	- -	- -	- -	$Dy_{10} + Dx_{10} + X \rightarrow Dx_{10}$ $-(Ay)_{10} + -(Ax)_{10} + X \rightarrow -(Ax)_{10}$	Add BCD source and eXtend bit to destination, BCD result
ADD [4]	BWL	s,Dn Dn,d	*****	e e	s d[4]	s d	s d	s d	s d	s d	s d	s -	s -	s -	s[4] -	s + Dn → Dn Dn + d → d	Add binary (ADDI or ADDQ is used when 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 -(Ay),-(Ax)	*****	e -	- -	- -	- -	- e	- -	- -	- -	- -	- -	- -	- -	Dy + Dx + X → Dx -(Ay) + -(Ax) + X → -(Ax)	Add source and eXtend bit to destination
AND [4]	BWL	s,Dn Dn,d	-**00	e e	- -	s d	s d	s d	s d	s d	s d	s -	s -	s -	s[4] -	s AND Dn → Dn Dn AND d → d	Logical AND source to destination (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 ASR	BWL W	Dx,Dy #n,Dy d	*****	e d -	- - -	- - d	- - d	- - d	- - d	- - d	- - d	- - d	- - -	- - -	- s -		Arithmetic shift Dy by Dx bits left/right Arithmetic shift Dy #n bits L/R (#n: 1 to 8) Arithmetic shift ds 1 bit left/right (.W only)
Bcc	BW[3]	address[2]	------	-	-	-	-	-	-	-	-	-	-	-	-	if cc true then address → PC	Branch conditionally (cc table on back) (8 or 16-bit ± offset to address)
BCHG	B L	Dn,d #n,d	--*--	e[1] d[1]	- -	d d	d d	d d	d d	d d	d d	d d	- -	- -	- s	NOT(bit number of d) → Z NOT(bit n of d)→ bit n of d	Set Z with state of specified bit in d then invert the bit in d
BCLR	B L	Dn,d #n,d	--*--	e[1] d[1]	- -	d d	d d	d d	d d	d d	d d	d d	- -	- -	- s	NOT(bit number of d) → Z 0 → bit number of d	Set Z with state of specified bit in d then 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 #n,d	--*--	e[1] d[1]	- -	d d	d d	d d	d d	d d	d d	d d	- -	- -	- s	NOT(bit n of d) → Z 1 → bit n of d	Set Z with state of specified bit in d then 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 #n,d	--*--	e[1] d[1]	- -	d d	d d	d d	d d	d d	d d	d d	d d	d d	- s	NOT(bit Dn of d) → Z NOT(bit #n of d) → Z	Set Z with state of specified bit in d 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,address[2]	------	-	-	-	-	-	-	-	-	-	-	-	-	if cc false then { Dn-1 → Dn if Dn <> -1 then addr →PC }	Test condition, decrement and branch (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; SP + #n → SP	Create local workspace on stack (negative n to allocate space)
LSL LSR	BWL W	Dx,Dy #n,Dy d	***0*	e d -	- - -	- - d	- - d	- - d	- - d	- - d	- - d	- - d	- - -	- - -	- s -		Logical shift Dy, Dx bits left/right Logical shift Dy, #n bits L/R (#n: 1 to 8) 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 An,USP	------	- -	d s	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	USP → An An → USP	Move User Stack Pointer to An (Privileged) 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 → Dx_{10}	Subtract BCD source and eXtend bit from
		-(Ay),-(Ax)		-	-	-	-	e	-	-	-	-	-	-	-	-(Ax)$_{10}$ - -(Ay)$_{10}$ – X →-(Ax)$_{10}$	destination, BCD result
Scc	B	d	————	d	-	d	d	d	d	d	d	d	-	-	-	If cc is true then 1's → d	If cc true then d.B = 11111111
																else 0's → d	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);	Push PC and SR, PC set by vector table #n
																(vector table entry) → PC	(#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; [u] Unsigned, [a] Alternate cc)					
cc	Condition	Test	cc	Condition	Test
T	true	1	VC	overflow clear	!V
F	false	0	VS	overflow set	V
HI[u]	higher than	!(C + Z)	PL	plus	!N
LS[u]	lower or same	C + Z	MI	minus	N
HS[u], CC[a]	higher or same	!C	GE	greater or equal	!(N ⊕ V)
LO[u], CS[a]	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

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

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

Distributed under the GNU general public use license.

17