

ALGO
QCM

1. Un sous-graphe G' de $G = \langle S, A \rangle$ est défini par ?
 - (a) $\langle S, A' \rangle$ avec $A' \subseteq A$
 - (b) $\langle S', A \rangle$ avec $S' \subseteq S$
 - (c) $\langle A, S \rangle$
2. Dans un graphe orienté, s'il existe un circuit $x \rightsquigarrow x$ passant par tous les sommets, le graphe est ?
 - (a) complet
 - (b) transitif
 - (c) connexe
 - (d) fortement connexe
3. Un graphe orienté de n sommets peut être fortement connexe à partir de ?
 - (a) $n - 1$ arcs
 - (b) n arcs
 - (c) $n + 1$ arcs
4. Deux sommets x et y d'un graphe orienté sont dits adjacents si ?
 - (a) il existe un arc $x \rightarrow y$ ou un arc $y \rightarrow x$
 - (b) il existe un arc $x \rightarrow y$ et un arc $y \rightarrow x$
 - (c) il existe un chemin $x \rightsquigarrow y$ ou un chemin $y \rightsquigarrow x$
 - (d) il existe un chemin $x \rightsquigarrow y$ et un chemin $y \rightsquigarrow x$
5. Une chaîne qui ne contient pas plusieurs fois un même sommet est ?
 - (a) élémentaire
 - (b) optimal
 - (c) plus court
 - (d) un chemin
6. Dans la forêt couvrante associée au parcours en profondeur d'un graphe orienté G , les arcs $x \rightarrow y$ tels que x est le père de y sont appelés ?
 - (a) Arcs couvrants
 - (b) Arcs en arrière
 - (c) Arcs en Avant
 - (d) Arcs croisés
7. Soit un graphe G connexe, sa fermeture transitive est ?
 - (a) Un sous-graphe
 - (b) Un graphe partiel
 - (c) Un graphe complet

8. L'algorithme de Warshall est utilisable sur ?
- (a) Les graphes orientés statiques
 - (b) Les graphes non orientés statiques
 - (c) Les graphes orientés évolutifs
 - (d) Les graphes non orientés évolutifs
9. Supposons que $Pref[i]$ retourne le Numéro d'ordre préfixe de rencontre d'un sommet i . Lors du parcours en profondeur d'un graphe orienté G , les arcs $x \rightarrow y$ tels que $pref[y]$ est inférieur à $Pref[x]$ dans la forêt sont appelés ?
- (a) Arcs couvrants
 - (b) Arcs en arrière
 - (c) Arcs en Avant
 - (d) Arcs croisés
10. Calculer la fermeture transitive d'un graphe sert à ?
- (a) Déterminer si un graphe est connexe
 - (b) Déterminer les composantes connexes d'un graphe non orienté
 - (c) Déterminer si un graphe est complet



QCM N°6

lundi 29 novembre 2021

Question 11

Dans l'espace vectoriel $E = \mathbb{R}^3$, considérons la famille $\mathcal{F} = \{(1, 1, 0), (2, -1, 0), (1, 1, -1)\}$.

- a. Cette famille est libre
- b. Cette famille est génératrice de \mathbb{R}^3
- c. Aucun des autres choix

Question 12

Dans \mathbb{R}^2 , considérons la base canonique $\mathcal{B} = \{(1, 0), (0, 1)\}$ et une autre base $\mathcal{B}' = \{(1, 2), (3, 4)\}$.

Pour tout vecteur $u = (x, y) \in \mathbb{R}^2$, on note $X = \begin{pmatrix} x \\ y \end{pmatrix}$ et $X' = \begin{pmatrix} x' \\ y' \end{pmatrix}$ ses coordonnées dans \mathcal{B} et \mathcal{B}' .

- a. $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$
- b. $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$
- c. La matrice de passage de \mathcal{B} à \mathcal{B}' est : $P = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$
- d. La matrice de passage de \mathcal{B} à \mathcal{B}' est : $P = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$
- e. Aucun des autres choix

Question 13

Parmi ces applications, lesquelles(laquelle) sont linéaire(s) ?

- a. $F : \begin{cases} \mathbb{R}^2 & \longrightarrow \mathbb{R} \\ (x, y) & \longmapsto x + 2y + xy \end{cases}$
- b. $G : \begin{cases} \mathbb{R}[X] & \longrightarrow \mathbb{R}[X] \\ P & \longmapsto P(1) + (X - 1)P'(1) + \frac{(X - 1)^2}{2}P''(1) \end{cases}$
- c. $H : \begin{cases} C^0(\mathbb{R}) & \longrightarrow \mathbb{R} \\ f & \longmapsto \int_0^1 x^2 f(x) dx \end{cases}$
- d. $K : \begin{cases} \mathbb{R}^{\mathbb{N}} & \longrightarrow \mathbb{R}^{\mathbb{N}} \\ (u_n) & \longmapsto ((n + 1)u_{n+1} - nu_n) \end{cases}$
- e. Aucune d'entre elles

Question 14

Soit $f \in \mathcal{L}(\mathbb{R}^3, \mathbb{R}^2)$ définie par sa matrice A dans les bases canoniques : $A = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \end{pmatrix}$

- a. $(1, -2) \in \text{Ker}(f)$
- b. $(1, -2) \in \text{Im}(f)$
- c. $(1, -2, 1) \in \text{Ker}(f)$
- d. $(1, -2, 1) \in \text{Im}(f)$
- e. Aucun des autres choix

Question 15

Soit la matrice $A = \begin{pmatrix} 1 & 0 & 2 \\ 1 & 1 & 1 \\ -1 & 0 & 1 \end{pmatrix}$.

- a. $\det(A) = 0$
- b. $\det(A) = 1$
- c. $\det(A) = 2$
- d. A est inversible
- e. Aucun des autres choix

Question 16

Soit $f \in \mathcal{L}(\mathbb{R}^3, \mathbb{R}^2)$ définie par sa matrice A dans les bases canoniques : $A = \begin{pmatrix} 1 & 1 & 3 \\ 1 & 2 & 1 \end{pmatrix}$

- a. Le rang de f vaut 1
- b. Le rang de f vaut 2
- c. Le rang de f vaut 3
- d. $\dim \text{Ker } f = 2$
- e. Aucun des autres choix

Question 17

Soient E un espace vectoriel sur \mathbb{R} , $f \in \mathcal{L}(E)$ et $\lambda \in \mathbb{R}$.

Le réel λ est valeur propre de f si et seulement si :

- a. $\forall u \in E, f(u) = \lambda u$
- b. $\forall u \in E, f(\lambda u) = u$
- c. $\exists u \in E$ tel que $u \neq 0_E$ et $f(\lambda u) = u$
- d. $\exists u \in E$ tel que $u \neq 0_E$ et $f(u) = \lambda u$
- e. Aucun des autres choix

Question 18

Soit la matrice : $A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 1 & 3 \end{pmatrix}$

Son polynôme caractéristique est :

- a. $P_A(X) = (1 - X)(2 - X)(3 - X)$
- b. $P_A(X) = (1 - X)[(2 - X)(3 - X) - 1]$
- c. $P_A(X) = (1 - X)[(2 - X)(3 - X) + 1]$
- d. Aucun des autres choix

Question 19

Soit $f \in \mathcal{L}(\mathbb{R}^2)$ définie par sa matrice dans la base canonique : $A = \begin{pmatrix} 5 & -2 \\ 3 & 0 \end{pmatrix}$.

Considérons les vecteurs $\varepsilon_1 = (2, 3)$ et $\varepsilon_2 = (1, 1)$.

- a. ε_1 et ε_2 sont des vecteurs propres de f .
- b. $\{\varepsilon_1, \varepsilon_2\}$ est une base de \mathbb{R}^2 .
- c. Pour tout $(a, b) \in \mathbb{R}^2$, $f(a\varepsilon_1 + b\varepsilon_2) = 2a\varepsilon_1 + 3b\varepsilon_2$
- d. Pour tout $(a, b) \in \mathbb{R}^2$, $f \circ f \circ f(a\varepsilon_1 + b\varepsilon_2) = 2^3 a\varepsilon_1 + 3^3 b\varepsilon_2$
- e. Aucun des autres choix

Question 20

Soit $f \in \mathcal{L}(\mathbb{R}^2)$ défini par sa matrice dans la base canonique : $A = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$

Considérons les vecteurs $\varepsilon_1 = (-2, 1)$ et $\varepsilon_2 = (1, 1)$, et la base de \mathbb{R}^2 : $\{\varepsilon_1, \varepsilon_2\}$.

Dans cette base, la matrice de f est :

a. $A' = \begin{pmatrix} 0 & 0 \\ 0 & 3 \end{pmatrix}$

b. $A' = \begin{pmatrix} 0 & 3 \\ 0 & 3 \end{pmatrix}$

c. Aucun des autres choix

Choose the one correct answer.

21. My friend won't ever lend me his car. I wish he _____ me his car for my date tomorrow night.
- would lend
 - will lend
 - was going to lend
 - lent
22. Jean Pierre did not come to the meeting. I wish _____
- he had come.
 - he has come.
 - he would have come.
 - he came.
23. The teacher is going to give us an exam tomorrow. I wish she _____ us an exam tomorrow.
- isn't going to give
 - won't give
 - weren't going to give
 - None of the above.

Numbers 24 and 25 are part of the same conversation.

24. Speaker A: I wish you ____! We're going to be late if you don't!
- hurry
 - will hurry
 - hurried
 - would hurry
25. Speaker B: I wish you _____. We have plenty of time.
- will relax
 - relax
 - would relax
 - were relaxing

Nos. 26 and 27 are part of the same situation.

26. Anne: How do you like the new president of our music association?
Karim: Not much. I wish she _____. I never should have voted for her.
- had not elected
 - had not been elected
 - would not have been elected
 - was not elected

27. Anne: Oh really? Then you probably wish I _____ for her either. If you recall, she won by only one vote. You and I could have changed the outcome of the election if we'd known then what we know now.

- a. have not voted
- b. didn't vote
- c. weren't voting
- d. had not voted

Choose the one correct auxiliary verb for these sentences.

28. I didn't read that book but I wish I _____.

- a. will
- b. had
- c. will have
- d. did

29. I have roommates, but I wish I _____ because I don't like them.

- a. did
- b. don't
- c. had not
- d. didn't

30. Objeet isn't old enough to drive a car, but he wishes he _____.

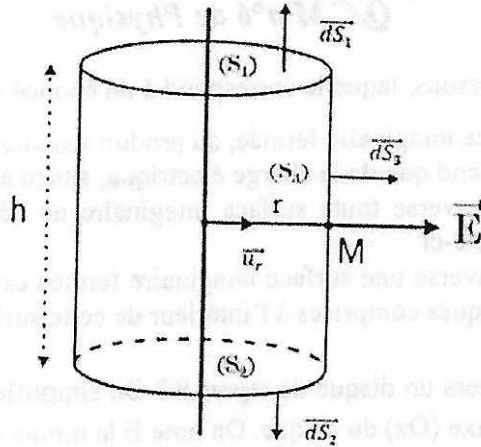
- a. were
- b. is
- c. would be
- d. None of the above.

OC : Q31 a 40 ⇒ cassettes video - merci.

Q.C.M n°6 de Physique

- 41- Parmi les propositions ci-dessous, laquelle correspond à un énoncé valable du théorème de Gauss :
- L'intégrale sur une surface imaginaire fermée, du produit scalaire du champ électrique par la surface élémentaire orientée ne dépend que de la charge électrique, située à l'extérieur de cette surface.
 - Le flux électrique qui traverse toute surface imaginaire ne dépend que de la charge électrique présente au voisinage de celle-ci
 - Le flux électrique qui traverse une surface imaginaire fermée est fonction de la somme algébrique de toutes les charges électriques comprises à l'intérieur de cette surface.
- 42- Que vaut le flux de \vec{E} à travers un disque de rayon R ? On simplifiera en prenant un champ uniforme qui forme un angle α avec l'axe (Oz) du disque. On note E la norme du vecteur \vec{E} .
- $\phi(\vec{E}) = \pi R^2 \cdot E \cdot \cos(\alpha)$
 - $\phi(\vec{E}) = 4\pi R^2 \cdot E \cdot \cos(\alpha)$
 - $\phi(\vec{E}) = 2\pi R \cdot E \cdot \cos(\alpha)$
- 43- Soit un fil de longueur L est chargé avec une densité linéique λ supposée constante. Le fil est placé sur l'axe (Oz). La charge élémentaire dQ d'un élément de longueur dl du fil s'exprime par :
- $dQ = \lambda L$
 - $dQ = \lambda dz$
 - $dQ = \lambda dr \cdot dz$
- 44- On considère le système chargé de la question (43). Dans ce cas la densité linéique du fil est d'expression :
- $\lambda(z) = a \cdot z$ (a est une constante).
- La charge totale du fil est:
- $Q = a L$
 - $Q = a \frac{L^2}{2}$
 - $Q = a L^2$
 - $Q = 2a L$
- 45- On considère un champ électrique radial sortant et une surface de Gauss cylindrique S_G , passant par M, le flux de $\vec{E}(M)$ est
- nul à travers la surface de base de S_G
 - maximal et non nul à travers la surface de base de S_G
 - maximal et non nul à travers la surface de coupe de S_G
- 46- Pour toute surface S , le flux ϕ du champ \vec{E} est donné par:
- $\Phi(\vec{E}) = \iint_S \frac{\partial \vec{E}}{\partial t} \cdot \vec{dS}$
 - $\Phi(\vec{E}) = \iint_S \vec{E} \wedge \vec{dS}$
 - $\Phi(\vec{E}) = \iint_S \vec{E} \cdot \vec{dS}$
- 47- Le champ $\vec{E}(M)$, généré par un cylindre creux infini de rayon R chargé avec une densité surfacique σ uniforme, en un point M à l'intérieur du cylindre, est donné par:
- $\vec{E}(M) = \vec{0}$
 - $\vec{E}(M) = \frac{k\sigma R}{r} \vec{u}_r$
 - $\vec{E}(M) = \frac{k\sigma}{r^2} \vec{u}_r$

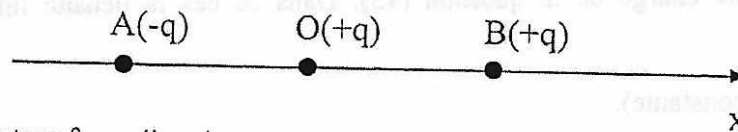
48- On considère un champ électrique radial et qui ne dépend que de la variable r , le flux électrique à travers le cylindre de rayon r et hauteur h , représenté sur la figure ci-dessous, s'écrit:



a) $\oint_S \vec{E} \cdot d\vec{S} = 2E \cdot S_1$ b) $\oint_S \vec{E} \cdot d\vec{S} = E \cdot \pi r^2 h$ c) $\oint_S \vec{E} \cdot d\vec{S} = E \cdot S_3$

49- Soit une distribution de charges ponctuelles représentée sur la figure ci-dessous :

($AB = 2a$ et O est milieu de AB).



La norme du vecteur force électrique exercée au point A s'exprime par :

a) $F(A) = \frac{2kq^2}{a^2}$ b) $F(A) = \frac{5kq^2}{4a^2}$ c) $F(A) = \frac{5kq}{4a^2}$ d) $F(A) = 0$

50- Une sphère de centre O et de rayon R est chargée en volume avec une densité de charges ρ positive et constante.

Les lignes de champ électrique $\vec{E}(M)$ engendrées par cette sphère sont

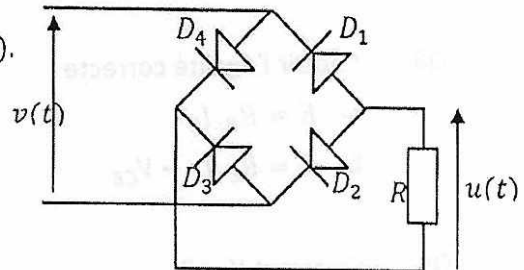
- a) non définies en $r = R$
- b) orthogonales à toute sphère de centre O et entrantes.
- c) orthogonales à toute sphère de centre O et sortantes.
- d) Fermées.

QCM Electronique – InfoS3

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

Q1. Soit le circuit suivant où $v(t) = V \cdot \sqrt{2} \cdot \sin(\omega t)$.
Choisir l'affirmation correcte :

- a- $u(t) \leq 0 \forall t$
- b- $u(t) \geq 0 \forall t$
- c- $u(t) = 0$ si $v(t) \leq 0$
- d- $u(t) = 0$ si $v(t) \geq 0$



Q2. En polarisation inverse, on peut représenter la diode Zéner à l'aide de l'un des 2 modèles : à seuil ou réel – le modèle idéal n'existant pas pour cette diode.

- a- VRAI
- b- FAUX

Si le gain en courant β d'un transistor bipolaire vaut 200 et le courant de base, $100\mu A$, alors en mode linéaire (Q3&4) :

Q3. le courant de collecteur vaut :

- a- $0,5mA$
- b- $20mA$
- c- $2mA$
- d- $0,5\mu A$

Q4. le courant d'émetteur vaut :

- a- $100,5mA$
- b- $2,1mA$
- c- $20,1mA$
- d- $100,5\mu A$

Q5. En mode normal (ou linéaire), la jonction base-émetteur est :

- a- Bloquée
- b- Passante

Q6. En mode saturé, que peut-on dire de la tension V_{CE} ?

- a- Elle est strictement positive
- b- Elle est quasiment nulle
- c- Elle est négative
- d- Elle dépend du reste du circuit

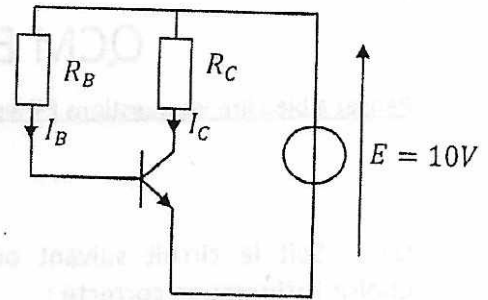
Q7. En mode saturé, que peut-on dire du courant I_B ?

- a- Il est supérieur à $\frac{I_C}{\beta}$
- b- Il est nul
- c- Il est strictement inférieur à $\frac{I_C}{\beta}$
- d- Il est strictement négatif

11.

Soit le circuit ci-contre (Q8 à 10):

On considère le cahier des charges suivant : $I_C = 20 \text{ mA}$, $V_{CE} = 5 \text{ V}$, et on prend un transistor ayant les caractéristiques suivantes : $\beta = 200$, $V_{BE} = 0,7 \text{ V}$ si la jonction Base-Emetteur est passante.



Q8. Choisir l'égalité correcte :

a- $E = R_B \cdot I_B$

b- $E = R_C \cdot I_C - V_{CE}$

c- $R_B \cdot I_B + V_{BC} = R_C \cdot I_C$

d- $V_{BC} = V_{BE} + V_{CE}$

Q9. Que vaut V_{BC} ? :

a- $0,6 \text{ V}$

b- $-5,7 \text{ V}$

c- $4,7 \text{ V}$

d- $-4,3 \text{ V}$

Q10. Que vaut R_C ? :

a- $2,5 \text{ k}\Omega$

b- 250Ω

c- 25Ω

d- 750Ω

QCM 6

Architecture des ordinateurs

Lundi 29 novembre 2021

Pour toutes les questions, une ou plusieurs réponses sont possibles.

11. Quelles instructions ne sont pas possibles ?

- A. MULS.W #\$80,D0
- B. MULU.W #80,D0
- C. MULS.L #\$80,D0
- D. MULU.L #80,D0

12. 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 :

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

13. Le *flag* C est positionné à 1 quand :

- A. Un dépassement signé apparaît.
- B. Un résultat est positif.
- C. Un dépassement non signé apparaît.
- D. Un résultat est négatif.

14. Quel(s) mnémonique(s) est (sont) une directive d'assemblage ?

- A. ORG
- B. MOVE
- C. ILLEGAL
- D. EQU

15. À quoi sert le symbole '#' ?

- A. Il indique qu'un opérande est sous forme hexadécimale.
- B. Il indique qu'un opérande est une donnée immédiate.
- C. Il indique qu'un opérande est sous forme décimale.
- D. Il indique qu'un opérande est une adresse.

16. Quels modes d'adressage ne spécifient pas d'emplacement mémoire ?

- A. Mode d'adressage immédiat.
- B. Mode d'adressage direct.
- C. Mode d'adressage indirect.
- D. Mode d'adressage absolu.

17. Quelle(s) instruction(s) peut-on utiliser pour appeler un sous-programme ?

- A. JMP
- B. Aucune de ces réponses.
- C. BRA
- D. BEQ

18. Les étapes pour empiler une donnée sont :

- A. Écrire la donnée dans (A7) puis décrémenter A7.
- B. Incrémenter A7 puis écrire la donnée dans (A7).
- C. Aucune de ces réponses.
- D. Décrémenter A7 puis écrire la donnée dans (A7).

19. Soient les deux instructions suivantes :

TST.W D0
BPL NEXT

L'instruction BPL effectue le branchement si :

- A. D0 = \$FFFFFF111
- B. D0 = \$88777788
- C. D0 = \$00000FFF
- D. D0 = \$FFFF1111

20. Soient les deux instructions suivantes :

CMP.B D1,D2
BLT NEXT

Si D2 = \$000000FF, l'instruction BLT effectue le branchement si :

- A. D1 = \$000000FE
- B. D1 = \$00000001
- C. D1 = \$FFFFFF01
- D. Aucune de ces réponses.

EASy68K Quick Reference v1.8

<http://www.wowgwap.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
ABCD	B	Dy.Dx -(Ay),-(Ax)	XNZVC	Dn An (An) (An)+ -(An) (iAn) (iAn,Rn) abs.W abs.L (iPC) (iPC,Rn) #n	-	$Dy_0 + Dx_0 + X \rightarrow Dx_0$ $-(Ay)_0 + -(Ax)_0 + X \rightarrow -(Ax)_0$	Add BCD source and extend bit to destination. BCD result
ADD	BWL	s.Dn Dn.d	*****	s s s s s s s s s s s s s s s s	s s s s s s s s s s s s s s s s	$s + Dn \rightarrow Dn$ $Dn + d \rightarrow d$	Add binary (ADDI or ADDQ is used when source is #n. Prevalent ADDQ with #n.L)
ADDA	WL	s.An	-----	s s s s s s s s s s s s s s s s	s s s s s s s s s s s s s s s s	$s + An \rightarrow An$	Add address (W sign-extended to L)
ADDI	BWL	#n.d	*****	d d d d d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	$\#n + d \rightarrow d$	Add immediate to destination
ADDQ	BWL	#n.d	*****	d d d d d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	$\#n + d \rightarrow d$	Add quick immediate (#n range: 1 to 8)
AOOX	BWL	Dy.Dx -(Ay),-(Ax)	*****	s s s s s s s s s s s s s s s s	s s s s s s s s s s s s s s s s	$Dy + Dx + X \rightarrow Dx$ $-(Ay) + -(Ax) + X \rightarrow -(Ax)$	Add source and extend bit to destination
AND	BWL	s.Dn Dn.d	---*00	s s s s s s s s s s s s s s s s	s s s s s s s s s s s s s s s s	s AND Dn \rightarrow Dn Dn AND d \rightarrow d	Logical AND source to destination (ANDI is used when source is #n)
ANDI	BWL	#n.d	---*00	d d d d d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	$\#n$ AND d \rightarrow d	Logical AND immediate to destination
ANDI	B	#n.CCR	=====	- - - - - - - - - - - - - - - -	s s s s s s s s s s s s s s s s	$\#n$ AND CCR \rightarrow CCR	Logical AND immediate to CCR
ANDI	W	#n.SR	=====	- - - - - - - - - - - - - - - -	s s s s s s s s s s s s s s s s	$\#n$ AND SR \rightarrow SR	Logical AND immediate to SR (Privileged)
ASL	BWL	Dx.Dy #n.Dy	*****	e d - - - - - - - - - - - - - -	s s s s s s s s s s s s s s s s	Arithmetic shift Dy by Dx bits left/right Arithmetic shift Dy #n bits L/R (#n: 1 to 8)	
ASR	W	d	- - - - -	- d d d d d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	Arithmetic shift ds 1 bit left/right (W only)	
Bcc	BW ³	address ²	-----	- - - - - - - - - - - - - - - -	- - - - - - - - - - - - - - - -	if cc true then address \rightarrow PC	Branch conditionally (cc table on back) (B or 16-bit \pm offset to address)
BCHG	B L	Dn.d #n.d	---*--	e ¹ d - d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	NOT(bit number of d) \rightarrow Z NOT(bit n of d) \rightarrow 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 - d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	NOT(bit number of d) \rightarrow Z 0 \rightarrow 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 \rightarrow PC	Branch always (B or 16-bit \pm offset to add)
BSET	B L	Dn.d #n.d	---*--	e ¹ d - d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	NOT(bit n of d) \rightarrow Z 1 \rightarrow 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 \rightarrow -(SP); address \rightarrow PC	Branch to subroutine (B or 16-bit \pm offset)
BTST	B L	Dn.d #n.d	---*--	e ¹ d - d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	NOT(bit Dn of d) \rightarrow Z NOT(bit #n of d) \rightarrow Z	Set Z with state of specified bit in d Leave the bit in d unchanged
CHK	W	s.Dn	---*00	e - s s s s s s s s s s s s s s	s s s s s s 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 d d d d d d d	s s s s s s s s s s s s s s s s	0 \rightarrow d	Clear destination to zero
CMP	BWL	s.Dn	---*--	e s ⁴ s s s s s s s s s s s s s s	s s s s s s s s s s s s s s s s	set CCR with Dn - s	Compare Dn to source
CMPA	WL	s.An	---*--	s s s s s s s s s s s s s s s s	s s s s s s s s s s s s s s s s	set CCR with An - s	Compare An to source
CMPI	BWL	#n.d	---*--	d - d d d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	set CCR with d - #n	Compare destination to #n
CMPW	BWL	(Ay)-(Ax)+	---*--	- - - - - - - - - - - - - - - -	- - - - - - - - - - - - - - - -	set CCR with (Ax) - (Ay)	Compare (Ax) to (Ay); increment Ax and Ay
DBcc	W	Dn.address ²	-----	- - - - - - - - - - - - - - - -	- - - - - - - - - - - - - - - -	if cc false then { Dn-1 \rightarrow Dn if Dn < -1 then addr \rightarrow PC }	Test condition, decrement and branch (16-bit \pm offset to address)
DIVS	W	s.Dn	---*00	e - s s s s s s s s s s s s s s	s s s s s s s s s s s s s s s s	$\pm 32\text{bit } Dn / \pm 16\text{bit } s \rightarrow \pm Dn$	Dn = (16-bit remainder, 16-bit quotient)
DIVU	W	s.Dn	---*00	e - s s s s s s s s s s s s s s	s s s s s s s s s s s s s s s s	$32\text{bit } Dn / 16\text{bit } s \rightarrow Dn$	Dn = (16-bit remainder, 16-bit quotient)
EOR	BWL	Dn.d	---*00	e - d d d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	Dn XOR d \rightarrow d	Logical exclusive OR Dn to destination
EORI	BWL	#n.d	---*00	d - d d d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	$\#n$ XOR d \rightarrow d	Logical exclusive OR #n to destination
EORI	B	#n.CCR	=====	- - - - - - - - - - - - - - - -	s s s s s s s s s s s s s s s s	$\#n$ XOR CCR \rightarrow CCR	Logical exclusive OR #n to CCR
EORI	W	#n.SR	=====	- - - - - - - - - - - - - - - -	s s s s s s s s s s s s s s s s	$\#n$ XOR SR \rightarrow SR	Logical exclusive OR #n to SR (Privileged)
EXG	L	Rx.Ry	-----	e s - - - - - - - - - - - - - -	- - - - - - - - - - - - - - - -	register \leftrightarrow register	Exchange registers (32-bit only)
EXT	WL	Dn	---*00	d - - - - - - - - - - - - - - -	- - - - - - - - - - - - - - - -	Dn.B \rightarrow Dn.W Dn.W \rightarrow Dn.L	Sign extend (change B to W or W to L)
ILLEGAL			-----	- - - - - - - - - - - - - - - -	- - - - - - - - - - - - - - - -	PC \rightarrow -(SSP); SR \rightarrow -(SSP)	Generate illegal instruction exception
JMP		d	-----	- - d - - - - - d d d d d d d d	- - - - - - - - - - - - - - - -	$\uparrow d \rightarrow$ PC	Jump to effective address of destination
JSR		d	-----	- - d - - - - - d d d d d d d d	- - - - - - - - - - - - - - - -	PC \rightarrow -(SP); $\uparrow d \rightarrow$ PC	push PC, jump to subroutine at address d
LEA	L	s.An	-----	- s s - - - - - s s s s s s s s	- - - - - - - - - - - - - - - -	$\uparrow s \rightarrow$ An	Load effective address of s to An
LINK		An.#n	-----	- - - - - - - - - - - - - - - -	- - - - - - - - - - - - - - - -	An \rightarrow -(SP); SP \rightarrow An; SP + #n \rightarrow SP	Create local workspace on stack (negative n to allocate space)
LSL	BWL	Dx.Dy #n.Dy	---*0*	e d - - - - - - - - - - - - - -	s s s s s s s s s s s s s s s s	Logical shift Dy, Dx bits left/right Logical shift Dy, #n bits L/R (#n: 1 to 8)	
LSR	W	d	- - - - -	- d d d d d d d d d d d d d d d d	s s s s s s s s s s s s s s s s	Logical shift d 1 bit left/right (W only)	
MOVE	BWL	s.d	---*00	e s ⁴ s s s s s s s s s s s s s s	s s s s s s s s s s s s s s s s	s \rightarrow d	Move data from source to destination
MOVE	W	s.CCR	=====	s - s s s s s s s s s s s s s s	s s s s s s s s s s s s s s s s	s \rightarrow CCR	Move source to Condition Code Register
MOVE	W	s.SR	=====	s - s s s s s s s s s s s s s s	s s s s s s s s s s s s s s s s	s \rightarrow SR	Move source to Status Register (Privileged)
MOVE	W	SR.d	-----	d - d d d d d d d d d d d d d d	- - - - - - - - - - - - - - - -	SR \rightarrow d	Move Status Register to destination
MOVE	L	USP.An An.USP	-----	- d - - - - - - - - - - - - - -	- - - - - - - - - - - - - - - -	USP \rightarrow An An \rightarrow USP	Move User Stack Pointer to An (Privileged) Move An to User Stack Pointer (Privileged)
	BWL	s.d	XNZVC	Dn An (An) (An)+ -(An) (iAn) (iAn,Rn) abs.W abs.L (iPC) (iPC,Rn) #n			

A'S

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

Condition Tests (+ OR, I NOT, ⊕ XOR; * Unsigned, * Alternate cc)					
cc	Condition	Test	cc	Condition	Test
T	true	I	VC	overflow clear	IV
F	false	D	VS	overflow set	V
H [*]	higher than	I(C + Z)	PL	plus	IN
LS [*]	lower or same	C + Z	MI	minus	N
HS [*] , CC [*]	higher or same	IC	GE	greater or equal	I(N ⊕ V)
LD [*] , CS [*]	lower than	C	LT	less than	(N ⊕ V)
NE	not equal	IZ	GT	greater than	I((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
a 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, D cleared, I set, U undefined

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

Distributed under the GNU general public use license.

16