

ALGO
QCM

1. Dans un graphe orienté, s'il existe un chemin $x \rightsquigarrow x$ passant par tous les sommets du graphe le graphe est ?
 - (a) complet
 - (b) partiel
 - (c) parfait
 - (d) fortement connexe
2. 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
3. Dans un graphe non orienté $G = \langle S, A \rangle$, Le sous-graphe connexe maximal $G' = \langle S', A \rangle$ est une composante connexe du graphe G ?
 - (a) vrai
 - (b) faux
4. Un graphe partiel 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) $\langle A, S \rangle$
5. Dans un graphe non orienté, s'il existe une arête $x - y$ pour tout couple de sommet $\{x, y\}$ le graphe est ?
 - (a) complet
 - (b) partiel
 - (c) parfait
 - (d) connexe
6. Dans un graphe orienté, on dit que l'arc $U = y \rightarrow x$ est ?
 - (a) incident à x vers l'extérieur
 - (b) accident à x vers l'extérieur
 - (c) incident à x vers l'intérieur
 - (d) accident à x vers l'intérieur

7. 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
8. Dans un graphe valué $G = \langle S, A, C \rangle$, les coûts sont portés par ?
- (a) les relations
 - (b) les sommets
9. Un chemin qui ne contient pas plusieurs fois un même sommet est ?
- (a) élémentaire
 - (b) optimal
 - (c) plus court
 - (d) une chaîne
10. Dans un graphe non orienté, une chaîne dont toutes les arêtes sont distinctes deux à deux et telle que les deux extrémités coïncident est ?
- (a) un circuit
 - (b) un cycle
 - (c) connexe
 - (d) fortement connexe
 - (e) un chemin



QCM N°5

lundi 22 novembre 2021

Question 11

Soit la série entière $\sum (-2)^n x^n$ et notons R son rayon de convergence.

Considérons sa fonction somme, définie pour tout $x \in]-R, R[$ par : $f(x) = \sum_{n=0}^{+\infty} (-2)^n x^n$

- a. Pour tout $x \in]-R, R[$, $f(x) = \frac{1}{1-2x}$
- b. Pour tout $x \in]-R, R[$, $f(x) = \frac{1}{1+2x}$
- c. Pour tout $x \in]-R, R[$, $f(x) = \frac{1}{1-\frac{x}{2}}$
- d. Pour tout $x \in]-R, R[$, $f(x) = \frac{1}{1+\frac{x}{2}}$
- e. Aucun des autres choix

Question 12

Soient $q \in]0, 1[$ et X une variable aléatoire entière telle que $G_X(t) = \frac{(1-q)}{1-qt}$

- a. $P(X=2) = (1-q)$
- b. $P(X=2) = (1-q) \times q$
- c. $P(X=2) = (1-q) \times q^2$
- d. Aucun des autres choix

Question 13

Soient E un espace vectoriel sur \mathbb{R} et $\mathcal{F} = \{e_1, \dots, e_n\}$ une famille de E .

\mathcal{F} est une famille génératrice de E si et seulement si :

- a. $\forall x \in E, \exists (\lambda_1, \dots, \lambda_n) \in \mathbb{R}^n$ tel que $x = \lambda_1 e_1 + \dots + \lambda_n e_n$
- b. $\exists x \in E, \forall (\lambda_1, \dots, \lambda_n) \in \mathbb{R}^n, x = \lambda_1 e_1 + \dots + \lambda_n e_n$
- c. $\forall (\lambda_1, \dots, \lambda_n) \in \mathbb{R}^n, \lambda_1 e_1 + \dots + \lambda_n e_n = 0_E \implies \lambda_1 = \dots = \lambda_n = 0$
- d. Aucun des autres choix

Question 14

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

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

Question 15

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. $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$
- d. $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$
- e. Aucun des autres choix.

Question 16

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

- a. $F : \begin{cases} \mathbb{R}^2 & \longrightarrow \mathbb{R} \\ (x, y) & \longmapsto x + 2y - 1 \end{cases}$
- b. $G : \begin{cases} \mathbb{R}[X] & \longrightarrow \mathbb{R}[X] \\ P & \longmapsto P(1) + (X - 1)P'(1) \end{cases}$
- c. $H : \begin{cases} C^0(\mathbb{R}) & \longrightarrow \mathbb{R} \\ f & \longmapsto \int_0^1 x f(x) dx \end{cases}$
- d. $K : \begin{cases} \mathbb{R}^N & \longrightarrow \mathbb{R} \\ (u_n) & \longmapsto u_0 - 2u_1^2 \end{cases}$
- e. Aucune d'entre elles

Question 17

Considérons l'application linéaire $f : \begin{cases} \mathbb{R}^3 & \rightarrow \mathbb{R}^2 \\ (x, y, z) & \mapsto (x + y + z, x + y + z) \end{cases}$

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

Question 18

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

Son déterminant vaut :

- a. $\det(A) = 2$
- b. $\det(A) = -2$
- c. $\det(A) = 3$
- d. $\det(A) = -3$
- e. Aucun des autres choix

Question 19

Considérons deux matrices $A \in \mathcal{M}_3(\mathbb{R})$ et $B \in \mathcal{M}_3(\mathbb{R})$.

- a. Pour tout $\lambda \in \mathbb{R}$, $\det(\lambda A) = \lambda \det(A)$
- b. $\det(A + B) = \det(A) + \det(B)$
- c. $\det(A \times B) = \det(A) \times \det(B)$
- d. Aucun des autres choix

Question 20

Soit Δ l'endomorphisme sur $\mathbb{R}^{\mathbb{N}}$ définie pour tout $(u_n) \in \mathbb{R}^{\mathbb{N}}$ par : $\Delta((u_n)) = (u_{n+1})$.

Considérons la suite $(u_n) = (2^n)$. Alors cette suite est vecteur propre de Δ , associé à la valeur propre $\lambda = 2$.

- a. Vrai
- b. Faux

Choose the **one** correct answer for each question.

21. Epita does not have air conditioning, but I wish it at least _____ some good fans.
- will have
 - would have
 - had
 - has
22. Afshari didn't come to the party on Saturday. I wish _____
- she had come.
 - she came.
 - she would have come.
 - she will come.
23. It's raining! I wish the sun _____ right now.
- shined
 - were shining
 - shines
 - would be shining
24. I don't know how to play the guitar. I wish I _____ how to play it!
- know
 - will know
 - had known
 - knew
25. Pedro forgot to write down his password. He wishes he _____ to write down his password.
- forgot
 - had forgotten
 - would not forget
 - had not forgotten
26. Susan didn't eat dinner before she went to bed. She wasn't hungry then, but she was at 2 in the morning. She wishes she _____ dinner.
- had eaten
 - had as usual
 - would have ate
 - ate
27. Declan did not come to Prologin. I wish he _____ to Prologin.
- had come
 - would come
 - come
 - came

28. Michael does not like being a truck driver. He wishes he ____ a taxicab driver instead.

- a. became
- b. had become
- c. has been
- d. would be

29. Michael wishes he ____ people around in his car instead of furniture.

- a. had driven
- b. drove
- c. has been driving
- d. drives

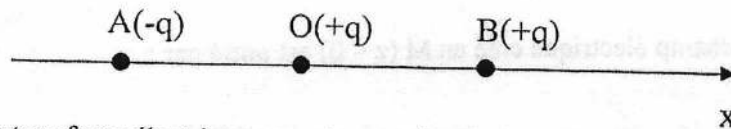
30. Maryam can't afford to come on holiday with us but I wish she _____, because I think we would have a great time together.

- a. will come
- b. can come
- c. would come
- d. None of the above.

Q.C.M n°5 de Physique

41- 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) $F(A) = 0$

42- On considère la distribution de charges de la question (41). Le potentiel électrique $V(O)$ créé au point O est

a) $V(O) = \frac{2k \cdot q}{a}$ b) $V(O) = 0$ c) $V(O) = -\frac{2k \cdot q}{a}$

43- Parmi les propositions ci-dessous, laquelle correspond à un énoncé valable du théorème de Gauss :

- a) 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.
- b) Le flux électrique qui traverse toute surface imaginaire ne dépend que de la charge électrique présente au voisinage de celle-ci
- c) 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.

44- 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} .

a) $\phi(\vec{E}) = \pi R^2 \cdot E \cdot \cos(\alpha)$ b) $\phi(\vec{E}) = 4\pi R^2 \cdot E \cdot \cos(\alpha)$ c) $\phi(\vec{E}) = 2\pi R \cdot E \cdot \cos(\alpha)$

45- Soit un anneau de rayon R et d'axe (Oz), chargé avec une densité linéique λ supposée constante. La charge élémentaire dQ d'un élément de longueur dl de l'anneau s'exprime par :

a) $dQ = \lambda d\theta$ b) $dQ = \lambda dR$ c) $dQ = \lambda R d\theta$

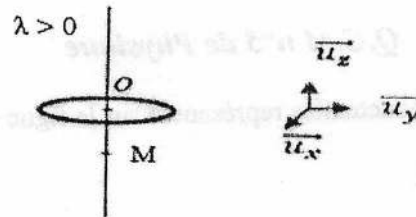
46- On considère le système chargé de la question (45). La charge totale de l'anneau est d'expression :

a) $Q = 2\pi R \lambda$ b) $Q = 2\pi \lambda$ c) $Q = \pi R \lambda$ d) $Q = \lambda R$

47- 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 :

- a) maximal et non nul à travers la surface de base de S_G
- b) nul à travers la surface de base de S_G
- c) maximal et non nul à travers la surface de coupe de S_G

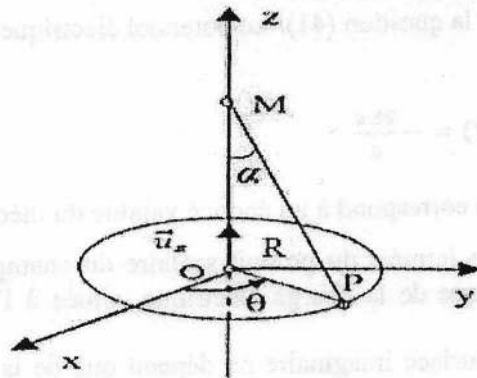
48- On considère un anneau dans le plan (x,y) centré en O et chargé uniformément avec une densité linéique $\lambda > 0$ (voir la figure ci-dessous).



Par symétrie, le vecteur champ électrique créé en M ($z < 0$) est porté par :

- a) $-\vec{u}_z$
- b) $+\vec{u}_z$
- c) Aucune des deux réponses précédentes n'est correcte

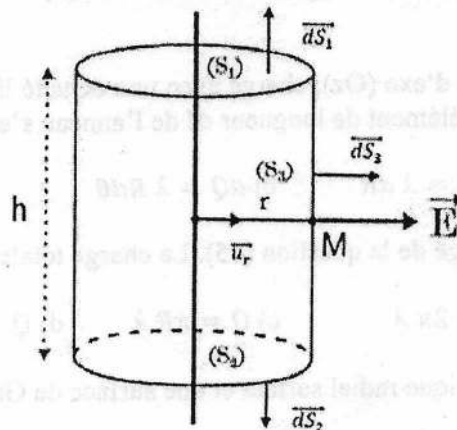
49- Un anneau de rayon R et d'axe (Oz) est chargé uniformément avec une densité linéique λ .



En un point M situé sur l'axe (Oz), le potentiel élémentaire est $dV(M) = \frac{k\lambda R d\theta}{PM}$; (P : point quelconque de l'anneau). On peut donc affirmer que le potentiel total créé par l'anneau au point M s'écrit :

- a) $V(z) = \frac{k\lambda R \pi}{\sqrt{z^2 + R^2}}$
- b) $V(z) = \frac{2k\lambda R \pi z}{\sqrt{z^2 + R^2}}$
- c) $V(z) = \frac{2k\lambda R \pi}{z^2 + R^2}$
- d) $V(z) = \frac{2k\lambda R \pi}{\sqrt{z^2 + R^2}}$

50- 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) $\oint_S \vec{E} \cdot d\vec{S} = E \cdot S_3$

QCM Electronique – InfoS3

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

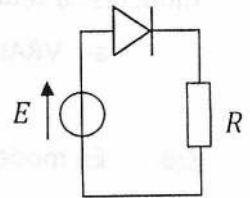
Q1. Soit le circuit ci-contre, dans lequel on considère la diode idéale. Que vaut la tension V_{AK} aux bornes de la diode si $E = 10V$, $R = 100\Omega$.

a- $10V$

c- $-10V$

b- $0V$

d- $0,7V$



Soit le circuit suivant où $v(t) = V \cdot \sqrt{2} \cdot \sin(\omega t)$. (Q2 à Q4)

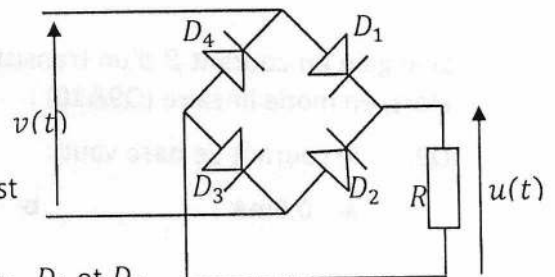
Q2. Quelles sont les diodes passantes si $v(t)$ est positive ?

a- D_1 et D_3

c- D_3 et D_4

b- D_2 et D_4

d- D_1 et D_2



Q3. Choisir l'affirmation correcte :

a- $u(t) \leq 0 \forall t$

c- $u(t) = 0$ si $v(t) \leq 0$

b- $u(t) \geq 0 \forall t$

d- $u(t) = 0$ si $v(t) \geq 0$

Q4. Que se passe-t-il si on modélise les diodes par leur modèle à seuil? On notera V_0 , la tension de seuil des diodes.

a- Si $|v| > 2 \cdot V_0$, alors les 4 diodes sont bloquées.

b- Si $|v| > V_0$, alors les 2 diodes de la question 2 sont passantes.

c- Si $|v| < 2 \cdot V_0$, alors les 4 diodes sont bloquées.

d- Toutes les réponses précédentes sont fausses.

Q5. Que se passe-t-il quand la tension appliquée aux bornes d'une diode devient très fortement négative (inférieure à une valeur spécifiée par le fabricant)

a- Il ne se passe rien

b- Le courant croît rapidement

c- Le courant décroît rapidement et il peut y avoir destruction de la diode.

d- Le courant croît puis devient nul.

11

- Q6. En polarisation directe, la diode Zéner se comporte comme un générateur de courant.
a- VRAI b- FAUX

- Q7. En polarisation inverse, 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

- Q8. En mode normal (ou linéaire), la jonction base-émetteur est :
a- Bloquée b- Passante

Si le gain en courant β d'un transistor bipolaire vaut 200 et le courant collecteur de 100mA, alors, en mode linéaire (Q9&10) :

- Q9. le courant de base vaut :
a- 0,5mA b- 0,2mA c- 2A d- 20A

- Q10. le courant d'émetteur vaut :
a- 100,5mA b- 100,2mA c- 2,1A d- 20,1A

QCM 5

Architecture des ordinateurs

Lundi 22 novembre 2021

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

11. 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.
12. Quelle(s) instruction(s) peut-on utiliser pour appeler un sous-programme ?
- A. JMP
 - B. BRA
 - C. Aucune de ces réponses.
 - D. BEQ
13. L'instruction BVS effectue un branchement si :
- A. C = 0
 - B. C = 1
 - C. V = 0
 - D. V = 1
14. Soient les deux instructions suivantes :
- ```
TST.W D0
BPL NEXT
```
- L'instruction BPL effectue le branchement si :
- A. D0 = \$00000FFF
  - B. D0 = \$FFFF1111
  - C. D0 = \$88777788
  - D. D0 = \$FFFFFF11
15. Soient les deux instructions suivantes :
- ```
CMP.B D1,D2  
BLT NEXT
```
- Si D2 = \$000000FF, l'instruction BLT effectue le branchement si :
- A. D1 = \$FFFFFF01
 - B. D1 = \$00000001
 - C. Aucune de ces réponses.
 - D. D1 = \$000000FE

16. Si $D0 = \$12345678$ et $D1 = \$87654321$, quelles sont les valeurs des *flags* après l'instruction suivante ?
ADD.B D0,D1

- A. $N = 1, Z = 0, V = 1, C = 1$
- B. $N = 1, Z = 0, V = 0, C = 1$
- C. $N = 1, Z = 0, V = 1, C = 0$
- D. Aucune de ces réponses.

17. Quelle instruction n'est pas possible ?

- A. ADDI.L #5,D0
- B. ADDQ.L #5,D2
- C. ADDI.L #30,D1
- D. ADDQ.L #30,D3

18. Quelles instructions ne sont pas possibles ?

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

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 :

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

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

14

EASy68K Quick Reference v1.8

<http://www.wowgwp.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	-	-	-	-	-	-	-	-	-	-	-	$Dy + Dx + X \rightarrow Dx$ $-(Ay)_{10} + -(Ax)_{10} + X \rightarrow -(Ax)_{10}$	Add BCD source and eXtend bit to destination. BCD result
ADD	BWL	s,Dn Dn,d	*****	e	s	d	s	s	s	s	s	s	s	s	s	$s + Dn \rightarrow Dn$ $Dn + d \rightarrow d$	Add binary (ADD or ADDU is used when source is #n. Prevent ADDQ with #n.L)
ADDA	WL	s,An	-----	s	e	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	-	-	s	$\#n + d \rightarrow d$	Add immediate to destination
ADDQ	BWL	#n,d	*****	d	d	d	d	d	d	d	d	d	-	-	s	$\#n + d \rightarrow d$	Add quick immediate (#n range: 1 to 8)
ADOX	BWL	Dy,Dx -(Ay),-(Ax)	*****	e	-	-	-	-	-	-	-	-	-	-	-	$Dy + Dx + X \rightarrow Dx$ $-(Ay) + -(Ax) + X \rightarrow -(Ax)$	Add source and eXtend bit to destination
AND	BWL	s,Dn Dn,d	---*00	e	-	s	s	s	s	s	s	s	s	s	s	$s \text{ AND } Dn \rightarrow Dn$ $Dn \text{ AND } d \rightarrow d$	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	-	-	s	$\#n \text{ AND } d \rightarrow d$	Logical AND immediate to destination
ANDI	B	#n,CCR	---*00	-	-	-	-	-	-	-	-	-	-	-	s	$\#n \text{ AND } CCR \rightarrow CCR$	Logical AND immediate to CCR
ANDI	W	#n,SR	---*00	-	-	-	-	-	-	-	-	-	-	-	s	$\#n \text{ AND } SR \rightarrow SR$	Logical AND immediate to SR (Privileged)
ASL	BWL	Dx,Dy #n,Dy	*****	e	-	-	-	-	-	-	-	-	-	-	s	$s \ll 1$ $s \ll \#n$	Arithmetic shift Dy by Dx bits left/right Arithmetic shift Dy #n bits L/R (#n: 1 to 8)
ASR	W	d	*****	d	-	-	-	-	-	-	-	-	-	-	s	$s \gg 1$ $s \gg \#n$	Arithmetic shift ds 1 bit left/right (W only)
Bcc	BW	address	-----	-	-	-	-	-	-	-	-	-	-	-	-	- 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	-	d	d	d	d	d	d	d	-	-	-	$\text{NOT}(\text{bit number of } d) \rightarrow Z$ $\text{NOT}(\text{bit } n \text{ of } d) \rightarrow \text{bit } n \text{ of } d$	Set Z with state of specified bit in d then invert the bit in d
BCLR	B L	Dn,d #n,d	---*--	e	-	d	d	d	d	d	d	d	-	-	-	$\text{NOT}(\text{bit number of } d) \rightarrow Z$ $0 \rightarrow \text{bit number of } d$	Set Z with state of specified bit in d then clear the bit in d
BRA	BW	address	-----	-	-	-	-	-	-	-	-	-	-	-	-	address \rightarrow PC	Branch always (B or 16-bit \pm offset to addr)
BSET	B L	Dn,d #n,d	---*--	e	-	d	d	d	d	d	d	d	-	-	-	$\text{NOT}(\text{bit } n \text{ of } d) \rightarrow Z$ $1 \rightarrow \text{bit } n \text{ of } d$	Set Z with state of specified bit in d then set the bit in d
BSR	BW	address	-----	-	-	-	-	-	-	-	-	-	-	-	-	PC \rightarrow -(SP); address \rightarrow PC	Branch to subroutine (B or 16-bit \pm offset)
BTST	B L	Dn,d #n,d	---*--	e	-	d	d	d	d	d	d	d	-	-	-	$\text{NOT}(\text{bit } Dn \text{ of } d) \rightarrow Z$ $\text{NOT}(\text{bit } \#n \text{ of } d) \rightarrow Z$	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	-	-	-	$0 \rightarrow d$	Clear destination to zero
CMP	BWL	s,Dn	-----	e	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	e	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	-	-	s	set CCR with $d - \#n$	Compare destination to #n
CMPI	BWL	(Ay),-(Ax)	-----	-	-	e	-	-	-	-	-	-	-	-	-	set CCR with $(Ax) - (Ay)$	Compare (Ax) to (Ay); Increment Ax and Ay
DBcc	W	Dn,address	-----	-	-	-	-	-	-	-	-	-	-	-	-	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	$\pm 32\text{bit } Dn / \pm 16\text{bit } s \rightarrow \pm Dn$	$Dn = [16\text{-bit remainder}, 16\text{-bit quotient}]$
DIVU	W	s,Dn	---*00	e	-	s	s	s	s	s	s	s	s	s	s	$32\text{bit } Dn / 16\text{bit } s \rightarrow Dn$	$Dn = [16\text{-bit remainder}, 16\text{-bit quotient}]$
EOR	BWL	Dn,d	---*00	e	-	d	d	d	d	d	d	d	-	-	s	$Dn \text{ XOR } d \rightarrow d$	Logical exclusive OR Dn to destination
EORI	BWL	#n,d	---*00	d	-	d	d	d	d	d	d	d	-	-	s	$\#n \text{ XOR } d \rightarrow d$	Logical exclusive OR #n to destination
EORI	B	#n,CCR	---*00	-	-	-	-	-	-	-	-	-	-	-	s	$\#n \text{ XOR } CCR \rightarrow CCR$	Logical exclusive OR #n to CCR
EORI	W	#n,SR	---*00	-	-	-	-	-	-	-	-	-	-	-	s	$\#n \text{ XOR } SR \rightarrow SR$	Logical exclusive OR #n to SR (Privileged)
EXG	L	Rx,Ry	-----	e	e	-	-	-	-	-	-	-	-	-	-	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	-	$Td \rightarrow PC$	Jump to effective address of destination
JSR		d	-----	-	-	d	-	-	d	d	d	d	d	d	-	PC \rightarrow -(SP); $Td \rightarrow PC$	push PC; jump to subroutine at address d
LEA	L	s,An	-----	-	e	s	-	-	s	s	s	s	s	s	-	$Ts \rightarrow An$	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	---*00	e	-	-	-	-	-	-	-	-	-	-	-	$s \ll 1$ $s \ll \#n$	Logical shift Dy, Dx bits left/right Logical shift Dy #n bits L/R (#n: 1 to 8)
LSR	W	d	---*00	d	-	-	-	-	-	-	-	-	-	-	s	$s \gg 1$ $s \gg \#n$	Logical shift d 1 bit left/right (W only)
MOVE	BWL	s,d	---*00	e	s	e	e	e	e	e	e	e	s	s	s	$s \rightarrow d$	Move data from source to destination
MOVE	W	s,CCR	---*00	s	-	s	s	s	s	s	s	s	s	s	s	$s \rightarrow CCR$	Move source to Condition Code Register
MOVE	W	s,SR	---*00	s	-	s	s	s	s	s	s	s	s	s	s	$s \rightarrow SR$	Move source to Status Register (Privileged)
MOVE	W	SR,d	---*00	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)	(i,An)	(i,An,Rn)	abs.W	abs.L	(i,PC)	(i,PC,Rn)	#n		

45

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)	(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	s → An	Move source to An (MOVE s,An use MOVEA)	
MOVEW ⁴	WL	Rn,Rn,d s,Rn-Rn	-----	-	-	d	-	d	d	d	d	d	-	-	-	-	Registers → d s → Registers	Move specified registers to/from memory (W source is sign-extended to L for Rn)	
MOVEP	WL	Dn,(iAn) (iAn),Dn	-----	s	-	-	-	-	d	-	-	-	-	-	-	-	Dn → (iAn)...(i+2An)...(i+4A. (iAn) → Dn...(i+2An)...(i+4A. (Access only even or odd addresses)		
MOVED 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	s	s16bit s * i16bit Dn → sDn	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	i16bit s * i16bit Dn → Dn	Multiply unisig'd 16-bit; result: unisig'd 32-bit	
NBCD	B	d	*U*U*	d	-	d	d	d	d	d	d	d	-	-	-	-	0 - d0 - 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			-----	-	-	-	-	-	-	-	-	-	-	-	-	-	-	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 Dn,d	---*00	e	-	s	s	s	s	s	s	s	s	s	s	s	s DR Dn → Dn Dn OR d → d	Logical OR (ORt 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	*00000	-	-	-	-	-	-	-	-	-	-	-	s	#n OR CCR → CCR	Logical OR #n to CCR		
ORI ⁴	W	#n,SR	*00000	-	-	-	-	-	-	-	-	-	-	-	s	#n OR SR → SR	Logical OR #n to SR (Privileged)		
PEA	L	s	-----	-	-	s	-	-	s	s	s	s	s	s	s	s	↑s → (SP)	Push effective address of s onto stack	
RESET			-----	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Assert RESET Line	
ROL	BWL	Dx,Dy	---*0*	e	-	-	-	-	-	-	-	-	-	-	-	-		Rotate Dx bits left/right (without X)	
ROR	BWL	#n,Dy	---*0*	d	-	-	-	-	-	-	-	-	-	-	s		Rotate Dy, #n bits left/right (#n: 1 to 8)		
ROXL	BWL	Dx,Dy	---*0*	e	-	-	-	-	-	-	-	-	-	-	-	-		Rotate Dx bits L/R, X used then updated	
ROXR	BWL	#n,Dy	---*0*	d	-	-	-	-	-	-	-	-	-	-	s		Rotate Dy, #n bits left/right (#n: 1 to 8)		
RTE			*00000	-	-	-	-	-	-	-	-	-	-	-	-	-	(SP)+ → SR; (SP)+ → PC	Return from exception (Privileged)	
RTR			*00000	-	-	-	-	-	-	-	-	-	-	-	-	-	(SP)+ → CCR; (SP)+ → PC	Return from subroutine and restore CCR	
RTS			-----	-	-	-	-	-	-	-	-	-	-	-	-	-	(SP)+ → PC	Return from subroutine	
SBCD	B	Dy,Dx -(Ay),-(Ax)	*U*U*	e	-	-	-	-	-	-	-	-	-	-	-	-	Dx0 - Dy0 - X → Dx0 -(Ax)0 - (Ay)0 - X → -(Ax)0	Subtract BCD source and eXtend bit from destination, BCD result	
Scc	B	d	-----	d	-	d	d	d	d	d	d	d	-	-	-	-	If cc true then 1's → d else 0's → d	If cc true then d = 11111111 else d = 00000000	
STOP ⁴		#n	*00000	-	-	-	-	-	-	-	-	-	-	-	s	#n → SR; STOP	Move #n to SR, stop processor (Privileged)		
SUB ⁴	BWL	s,Dn Dn,d	*****	e	s	s	s	s	s	s	s	s	s	s	s	s	Dn - s → Dn d - Dn → d	Subtract binary (SUBI or SUBQ used when 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	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	-	-	s	d - #n → d	Subtract quick immediate (#n range: 1 to 8)		
SUBX	BWL	Dy,Dx -(Ay),-(Ax)	*****	e	-	-	-	-	-	-	-	-	-	-	-	-	Dx - Dy - X → Dx -(Ax) - (Ay) - X → -(Ax)	Subtract source and eXtend bit from 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)	(iAn)	(iAn,Rn)	abs.W	abs.L	(iPC)	(iPC,Rn)	#n				

Condition Tests (* OR, 1 NOT, * XOR, * Unsigned, * Alternate cc)					
cc	Condition	Test	cc	Condition	Test
I	true	I	VC	overflow clear	IV
F	false	O	VS	overflow set	V
HI*	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)
LO*, 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

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, D 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.

16