

ALGO  
QCM

1. 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
2. Une chaîne qui ne contient pas plusieurs fois un même sommet est ?
  - (a) élémentaire
  - (b) optimal
  - (c) plus court
  - (d) un chemin
3. Soit un graphe  $G$  connexe, sa fermeture transitive est ?
  - (a) Un sous-graphe
  - (b) Un graphe partiel
  - (c) Un graphe complet
4. 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 supérieur à  $Pref[x]$  dans la forêt sont appelés ?
  - (a) Arcs couvrants
  - (b) Arcs en arrière
  - (c) Arcs en Avant
  - (d) Arcs croisés
5. 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
6. la longueur d'un chemin est ?
  - (a) éventuellement nulle.
  - (b) le nombre d'arcs qui le composent.
  - (c) le nombre de sommets qui le composent.
  - (d) le nombre d'arêtes qui le composent.

7. Un graphe non orienté de  $n$  sommets peut être connexe à partir de ?
- (a)  $n - 1$  arêtes.
  - (b)  $n$  arêtes.
  - (c)  $n + 1$  arêtes.
8. Pour déterminer les composantes connexes d'un graphe non orienté on peut utiliser ?
- (a) l'algorithme de *parcours en profondeur*.
  - (b) l'algorithme de *parcours en largeur*.
  - (c) l'algorithme de *Warshall*.
9. L'algorithme de *Warshall* permet de ?
- (a) calculer la fermeture transitive d'un graphe non orienté.
  - (b) calculer la fermeture transitive d'un graphe orienté.
  - (c) parcourir un graphe en largeur.
  - (d) déterminer si un graphe est complet.
10. Les algorithmes *Trouver* et *Réunir* nécessitent ?
- (a) un vecteur de pères.
  - (b) un vecteur de fils.
  - (c) un vecteur de frères.



## QCM N°1

Lundi 17 janvier 2022

### Question 11

Considérons l'intégrale  $I = \int_0^{\pi} x \sin(2x) dx$ . La formule d'intégration par parties donne :

a.  $I = [-x \cos(2x)]_0^{\pi} + \int_0^{\pi} \cos(2x) dx$

b.  $I = [x \cos(2x)]_0^{\pi} - \int_0^{\pi} \cos(2x) dx$

c.  $I = \left[ \frac{-x \cos(2x)}{2} \right]_0^{\pi} + \int_0^{\pi} \frac{\cos(2x)}{2} dx$

d.  $I = \left[ \frac{x \cos(2x)}{2} \right]_0^{\pi} - \int_0^{\pi} \frac{\cos(2x)}{2} dx$

e. Rien de ce qui précède

### Question 12

Considérons l'intégrale  $I = \int_0^1 \frac{1}{1+e^x} dx$ . Le changement de variable  $t = e^x$  donne :

a.  $I = \int_1^e \frac{1}{t(1+t)} dt$

b.  $I = \int_0^1 \frac{1}{t(1+t)} dt$

c.  $I = \int_1^e \frac{1}{1+t} dt$

d.  $I = \int_0^1 \frac{1}{1+t} dt$

e. Rien de ce qui précède

### Question 13

L'intégrale  $\int_0^1 \frac{1}{t^{\alpha}} dt$  converge si et seulement si :

a.  $\alpha < 0$

b.  $\alpha > 0$

c.  $\alpha < 1$

d.  $\alpha > 1$

e. Rien de ce qui précède

### Question 14

L'intégrale  $\int_1^{+\infty} \frac{1}{t^\alpha} dt$  converge si et seulement si :

- a.  $\alpha < 0$
- b.  $\alpha > 0$
- c.  $\alpha < 1$
- d.  $\alpha > 1$
- e. Rien de ce qui précède

### Question 15

Quelle est la nature de l'intégrale impropre  $\int_0^{+\infty} \sin(t) dt$  ?

- a. Convergente
- b. Divergente

### Question 16

Soit une fonction  $f$  continue sur  $\mathbb{R}^+$  telle que  $\int_0^{+\infty} f(t) dt$  converge.

Alors  $\lim_{x \rightarrow +\infty} \int_x^{2x} f(t) dt = 0$ .

- a. Vrai
- b. Faux

### Question 17

Soient deux fonctions  $f$  et  $g$  positives, continues sur  $]0, 3]$  et telles que pour tout  $t \in ]0, 3]$ ,  $0 \leq f(t) \leq g(t)$ .

Considérons les intégrales :  $I = \int_0^3 f(t) dt$  et  $J = \int_0^3 g(t) dt$ .

- a. Les intégrales  $I$  et  $J$  sont de même nature.
- b. Si  $I$  converge, alors  $J$  converge.
- c. Si  $I$  diverge, alors  $J$  diverge.
- d. Rien de ce qui précède

### Question 18

Soient deux fonctions  $f$  et  $g$  positives, continues sur  $]0, 3]$  et telles qu'au voisinage de 0,  $f(t) \sim g(t)$ .

Considérons les intégrales :  $I = \int_0^3 f(t) dt$  et  $J = \int_0^3 g(t) dt$ .

- a. Les intégrales  $I$  et  $J$  sont de même nature.
- b. Si  $I$  converge, alors  $J$  converge.
- c. Si  $I$  diverge, alors  $J$  diverge.
- d. Rien de ce qui précède

### Question 19

Quelle est la nature de l'intégrale  $\int_0^{\frac{\pi}{2}} \frac{1}{\sin(t)} dt$  ?

- a. Convergente
- b. Divergente

### Question 20

Soit une fonction  $f$  continue sur  $\mathbb{R}$ , telle que  $\int_{-\infty}^0 f(t) dt$  et  $\int_0^{+\infty} f(t) dt$  convergent toutes les deux.

- a.  $\int_1^{+\infty} f(t) dt$  converge
- b.  $\int_{-\infty}^1 f(t) dt$  converge
- c.  $\int_{-\infty}^{+\infty} f(t) dt$  converge
- d. Rien de ce qui précède

QCM 1 (conds1,2;wishegu)

21. Of course, you have not inherited a fortune. Which do you say?
- If I inherit a fortune, I buy EPITA.
  - If I inherited a fortune, I would buy EPITA.
  - If I had inherited a fortune, I would have bought EPITA.
  - If I inherit a fortune, I would buy EPITA.
22. You cannot go to the PSG game because it is too expensive. You say:
- I wish the tickets will be cheaper.
  - I wish the tickets were cheapest.
  - I wish the tickets are cheaper.
  - I wish the tickets were cheaper.
23. Somebody stops you on the way to EPITA to ask directions. You say:
- If you went right at the end of this street, you see EPITA on your left.
  - If you go right at the end of this street, you will see EPITA on your left.
  - If you had gone right at the end of this street, you will see EPITA on your left.
  - If you go right at the end of this street, you saw EPITA on your left.
24. You can't afford to buy a Mac. You say: "If I \_\_\_\_ a Mac, I would have to borrow the money."
- wanted to buy
  - buyed
  - would buy
  - had bought
25. If it rains a lot where you live, you say:
- I wish it didn't rain so much.
  - I wish it isn't raining so much.
  - I wish it wasn't raining so much.
  - I wish it doesn't rain so much.
26. You want to phone John but you don't have his number. Which sentence expresses this situation?
- I wish I know his number.
  - I wish I would know his number.
  - I wish I knew his number.
  - I wished I know his number.
27. Choose the sentence with no mistakes.
- If I am rich, I would buy a new car.
  - If I were rich, I would have buy a new car.
  - If I was rich, I would bought a new car.
  - If I were rich, I would buy a new car.

28. \_\_\_\_\_ the flight is cancelled, the seminar will have to be postponed.

- a. While
- b. If
- c. Although
- d. Besides

29. A father is not happy because his children generally don't obey him. He says to them:

- a. I wish you listen to me.
- b. I wish you will listen to me.
- c. I wish you listened to me.
- d. I wish you were listening to me.

30. Which of the following is the only correct sentence with no mistakes?

- a. I will buy the stock only if interested rates go down.
- b. I will have bought the stock only if interest rates go down.
- c. I will buy the stock only if interest rates goes down.
- d. I will buy the stock only if interest rates go down.

41. Deux charges égales, l'une au repos, l'autre ayant une vitesse de 10 m/s sont libérées dans un champ magnétique uniforme. Quelle charge a la plus grande force exercée sur elle par le champ magnétique ?

- La charge qui est au repos.
- La charge qui se déplace, si sa vitesse est parallèle à la direction du champ magnétique lorsqu'elle est relâchée.
- La charge qui se déplace si sa vitesse fait un angle de  $45^\circ$  avec la direction du champ magnétique lorsqu'elle est libérée.
- La charge qui se déplace si sa vitesse est perpendiculaire à la direction du champ magnétique lorsqu'elle est libérée.
- Toutes les charges ci-dessus subissent des forces égales lorsqu'elles sont libérées dans le même champ magnétique.

42- Une charge électrique  $q = 2\mu C$  se déplace avec une vitesse  $\vec{v} = \vec{u}_x - \vec{u}_z$  et subit un champ magnétique uniforme  $\vec{B} = (2\vec{u}_y - 3\vec{u}_z) \cdot 10^{-3}$ . On peut affirmer que la force magnétique s'exprimera par :

a)  $\vec{F} = 2 \cdot 10^{-9} (2\vec{u}_x - 3\vec{u}_y + 2\vec{u}_z)$

b)  $\vec{F} = 2 \cdot 10^{-9} (2\vec{u}_x + 3\vec{u}_y + 2\vec{u}_z)$

c)  $\vec{F} = 2 \cdot 10^{-6} (2\vec{u}_x - 3\vec{u}_y + 2\vec{u}_z)$

43- Considérons une charge électrique  $q = 1\mu C$  se déplaçant avec une vitesse  $v = 10$  m/s dans un champ magnétique uniforme de norme  $B = 1T$ . L'angle formé par les vecteurs vitesse et champ magnétique est  $\alpha = 90^\circ$ . La norme de la force magnétique est ?

a)  $F = 10^{-7} N$       b)  $F = 10^{-8} N$       c)  $F = 10^{-5} N$

44- Considérons une particule de charge  $q$  se déplaçant dans un champ magnétique  $\vec{B}$  uniforme orthogonal au vecteur vitesse de la particule.

Laquelle des affirmations suivantes est vraie ?

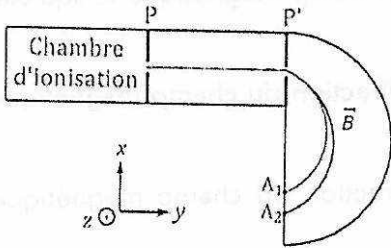
- La force magnétique force  $\vec{F}_m$  est nulle..
- Le travail de  $\vec{F}_m$  est strictement positif..
- La force magnétique  $\vec{F}_m$  est perpendiculaire au vecteur vitesse  $\vec{v}$



45- Une particule de charge positive  $q$  et de masse  $m$  se déplace dans un champ magnétique  $\vec{B}$  uniforme et perpendiculaire au vecteur vitesse  $\vec{v}$  de la particule . Le rayon de la trajectoire de la particule dans ce champ est :

a)  $R = \frac{mB}{qV}$       b)  $R = \frac{mV}{qB}$       c)  $R = \frac{qB}{mV}$       d)  $R = \frac{mq}{B}$

46- Des ions chargés positivement sont accélérés de P à P' au moyen d'un champ électrique puis déviés dans une région où règne de champ magnétique uniforme  $\vec{B}$  (voir figure ci-dessous). Quelle doit être la direction et le sens du vecteur  $\vec{B}$  pour que ces ions puissent arriver aux points  $A_1$  et  $A_2$  ?



a) Sur l'axe X, vers les  $x > 0$    b) Sur l'axe Z, vers les  $z > 0$    c) Sur l'axe Z, vers les  $z < 0$ .

47. Par définition, les lignes de champ du champ magnétique sont l'ensemble des courbes tangentes « en tout point » à  $\vec{B}$ .

a. VRAI

b. FAUX

48. Le vecteur de champ est tangent à la ligne de champ en tout point.

a.  $VRA_{it}$

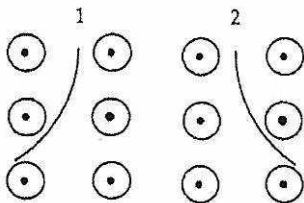
b. FAUX

49. Les lignes de champs relient les pôles magnétiques, et par convention sont orientées de sorte que les lignes de champ d'un aimant entrent par le nord (+) et ressortent par le sud (-).

a. VRAI

b. FAUX

50. Soit un champ magnétique sortant et orthogonal à la page. Deux particules chargées entrent par le haut et empruntent les chemins indiqués sur la figure ci-dessous. Quelle est l'affirmation correcte?



a. La particule 1 a une charge positive et la particule 2 a une charge négative

b. Les deux particules sont chargées positivement

c. Les deux particules sont chargées négativement

d. La particule 1 a une charge négative et la particule 2 a une charge positive.

e. La direction des chemins dépend de l'amplitude de la vitesse, pas du signe de la charge.

## QCM Electronique – InfoS4

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

Q1. Une résistance placée en série avec un générateur de courant modifie-t-elle l'intensité du courant délivré par ce générateur ?

a- OUI

b- NON

c- Ça dépend.

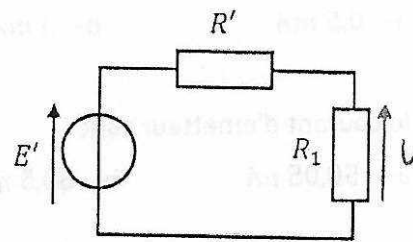
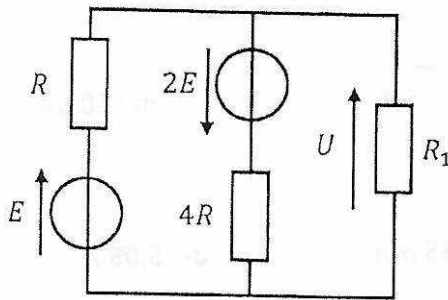
Q2. Quand on associe 2 résistances  $R_1$  et  $R_2$  en série, on conserve :

a. Le courant qui traverse  $R_2$

c. Rien du tout

b. la tension aux bornes de  $R_1$

Q3. On considère les 2 circuits suivants :



Ces 2 circuits sont équivalents si et seulement si :

a-  $E' = \frac{2}{5}E$  et  $R' = \frac{4}{5}R$

c-  $E' = -\frac{2}{5}E$  et  $R' = \frac{4}{5}R$

b-  $E' = \frac{6}{5}E$  et  $R' = \frac{4}{5}R$

d-  $E' = \frac{5}{2}E$  et  $R' = 5R$

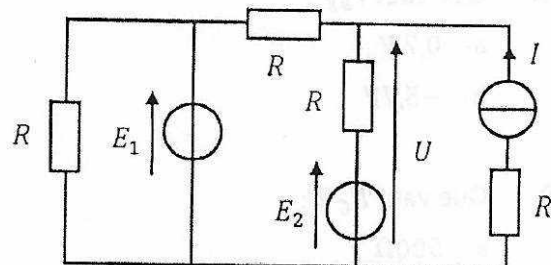
Q4. Quelle est la bonne formule ?

a.  $U = \frac{R \cdot I - E_1 + E_2}{3}$

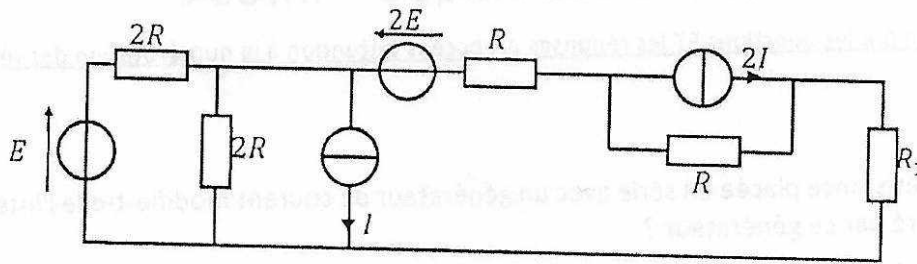
b.  $U = \frac{R \cdot I + E_1 + E_2}{2}$

c.  $U = \frac{R \cdot I + E_1 + E_2}{3}$

d.  $U = \frac{R \cdot I - E_1 + E_2}{4}$



Q5. Soit le montage ci-dessous.



Le générateur de Thévenin vu par  $R_1$  est tel que :

a-  $E_{th} = \frac{2RI - 3E}{2}$  et  $R_{th} = 3R$

c-  $E_{th} = \frac{6RI - 3E}{2}$  et  $R_{th} = 3R$

b-  $E_{th} = 2RI - 3E$  et  $R_{th} = 6R$

d-  $E_{th} = 6RI - 3E$  et  $R_{th} = 6R$

Si le gain en courant d'un transistor bipolaire est 100 et le courant collecteur de 50 mA, alors en fonctionnement linéaire :

Q6. le courant de base vaut :

a- 0,5 mA

b- 5 mA

c- 5 A

d- 50  $\mu$ A

Q7. le courant d'émetteur vaut :

a- 50,05  $\mu$ A

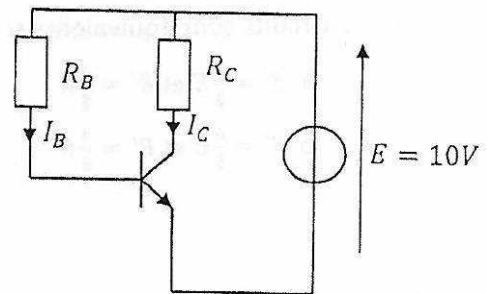
b- 50,5 mA

c- 55 mA

d- 5,05 A

Soit le circuit ci-contre (Q8&9):

On considère le cahier des charges suivant :  $I_C = 10$  mA,  $V_{CE} = 5$  V, et on prend un transistor ayant les caractéristiques suivantes :  $\beta = 200$ ,  $V_{BE} = 0,7$  V si la jonction Base-Emetteur est passante.



Q8. Que vaut  $V_{BE}$  ? :

a- 0,7 V

b- -5,7 V

c- 4,7 V

d- -4,3 V

Q9. Que vaut  $R_C$  ? :

a- 500  $\Omega$

b- 250  $\Omega$

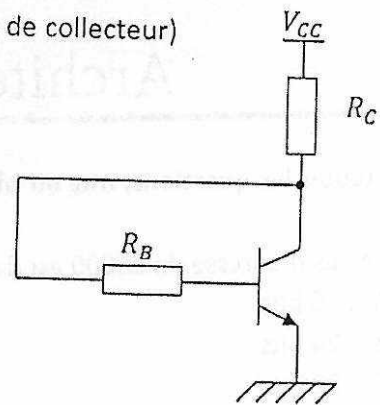
c- 25  $\Omega$

d- 750  $\Omega$

Q10. On considère le montage suivant (montage par réaction de collecteur)

Si on fait une loi des mailles, on trouve :

- a-  $V_{cc} = R_C \cdot I_C + R_B \cdot I_B + V_{BE}$
- b-  $V_{cc} = (\beta + 1) \cdot R_C \cdot I_B + R_B I_B + V_{BE}$
- c-  $V_{cc} = \beta \cdot R_C \cdot I_C + R_B I_B - V_{BE}$
- d-  $V_{cc} = \beta \cdot R_C \cdot I_B + R_B \cdot I_B$



# QCM 1

## Architecture des ordinateurs

Lundi 17 janvier 2022

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

11. Le bus d'adresse du 68000 est de :
  - A. 16 bits
  - B. 24 bits
  - C. 32 bits
  - D. 64 bits
12. Le 68000 possède :
  - A. 1 registre d'état
  - B. 2 registres d'état
  - C. 4 registres d'état
  - D. 8 registres d'état
13. Soit l'instruction suivante : `MOVE.W (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.
14. 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.
15. Les étapes pour dépiler une donnée sont :
  - A. Lire la donnée dans (A7) puis décrémenter A7.
  - B. Incrémenter A7 puis lire la donnée dans (A7).
  - C. Décrémenter A7 puis lire la donnée dans (A7).
  - D. Aucune de ces réponses.
16. L'instruction `RTS` :
  - A. Ne modifie pas la pile.
  - B. Empile une adresse de retour.
  - C. Restaure les registres.
  - D. Aucune de ces réponses.

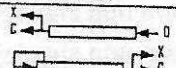
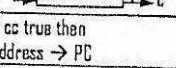

17. L'instruction BCC effectue un branchement si :
- A.  $C = 0$
  - B.  $C = 1$
  - C.  $V = 0$
  - D.  $V = 1$
18. Soient les deux instructions suivantes :
- ```
CMP.L D1,D2
BHI  NEXT
```
- L'instruction BHI effectue le branchement si :
- A.  $D2 > D1$  (comparaison signée)
  - B.  $D1 > D2$  (comparaison signée)
  - C.  $D1 > D2$  (comparaison non signée)
  - D.  $D2 > D1$  (comparaison non signée)
19. Si  $D0 = \$12345678$  et  $D1 = \$87654321$ , quelles sont les valeurs des *flags* après l'instruction suivante ?
- ```
ADD.B D0,D1
```
- A.  $N = 1, Z = 0, V = 1, C = 0$
  - B.  $N = 1, Z = 0, V = 1, C = 1$
  - C.  $N = 1, Z = 0, V = 0, C = 1$
  - D. Aucune de ces réponses.
20. Quelle instruction n'est pas possible ?
- A. `ADDI.L #5,D0`
  - B. `ADDQ.L #5,D2`
  - C. `ADDQ.L #30,D3`
  - D. `ADDI.L #30,D1`

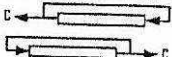
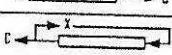
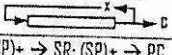
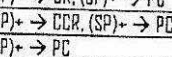


EASy68K Quick Reference v1.8

<http://www.wowgweb.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
		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*	B	-	-	-	-	-	-	-	-	-	-	-	$Dy + Dx + X \rightarrow Dx$ $-(Ay)_0 + -(Ax)_0 + X \rightarrow -(Ax)_0$	Add BCD source and eXtend bit to destination, BCD result	
ADD	BWL	s,Dn Dn,d	*****	B	s	s	s	s	s	s	s	s	s	s	s	$s + Dn \rightarrow Dn$ $Dn + d \rightarrow d$	Add binary (ADD or ADDQ is used when source is #n. Prevent ADDQ with #n.L)	
ADDA	WL	s,An	-----	B	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	-	-	s	$#n + d \rightarrow d$	Add immediate (W sign-extended to L)	
ADDQ	BWL	#n,d	*****	d	d	d	d	d	d	d	d	d	-	-	s	$#n + d \rightarrow d$	Add immediate to destination	
ADDX	BWL	Dy,Dx -(Ay),-(Ax)	*****	B	-	-	-	-	-	-	-	-	-	-	-	$Dy + Dx + X \rightarrow Dx$ $-(Ay) + -(Ax) + X \rightarrow -(Ax)$	Add source and eXtend bit to destination	
AND	BWL	s,Dn Dn,d	---*00	B	-	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	-----	-	-	-	-	-	-	-	-	-	-	-	-	$#n \text{ AND } CCR \rightarrow CCR$	Logical AND immediate to CCR	
ANDI	W	#n,SR	-----	-	-	-	-	-	-	-	-	-	-	-	-	$#n \text{ AND } SR \rightarrow SR$	Logical AND immediate to SR (Privileged)	
ASL	BWL	Dx,Dy #n,Dy	*****	B	-	-	-	-	-	-	-	-	-	-	-		Arithmetic shift Dy by Dx bits left/right	
ASR	W	d	-----	d	-	d	d	d	d	d	d	d	-	-	s		Arithmetic shift Dy #n bits L/R (#n: 1 to 8)	
Bcc	BW	address <sup>2</sup>	-----	-	-	-	-	-	-	-	-	-	-	-	-	- if cc true then address $\rightarrow$ PC	Branch conditionally (cc table on back) (8 or 16-bit $\pm$ offset to address)	
BCHG	B L	Dn,d #n,d	---*--	B	-	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	---*--	B	-	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 <sup>2</sup>	-----	-	-	-	-	-	-	-	-	-	-	-	-	address $\rightarrow$ PC	Branch always (8 or 16-bit $\pm$ offset to address)	
BSET	B L	Dn,d #n,d	---*--	B	-	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 <sup>2</sup>	-----	-	-	-	-	-	-	-	-	-	-	-	-	PC $\rightarrow$ -(SP); address $\rightarrow$ PC	Branch to subroutine (8 or 16-bit $\pm$ offset)	
BTST	B L	Dn,d #n,d	---*--	B	-	d	d	d	d	d	d	d	d	d	s	$\text{NOT}(\text{bit } 0 \text{ of } d) \rightarrow Z$ $\text{NOT}(\text{bit } n \text{ of } d) \rightarrow Z$	Set Z with state of specified bit in d Leave the bit in d unchanged	
CHK	W	s,Dn	---UUU	B	-	s	s	s	s	s	s	s	s	s	s	if $0 \leq Dn < Dn$ 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	
CMPI	BWL	s,Dn	---***	B	s	s	s	s	s	s	s	s	s	s	s	set CCR with $Dn - s$	Compare Dn to source	
CMPI	WL	s,An	---***	B	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	-	-	s	set CCR with $d - \#n$	Compare destination to #n	
CMPI	BWL	(Ay),-(Ax)	---***	-	-	-	-	-	-	-	-	-	-	-	-	set CCR with $(Ax) - (Ay)$	Compare (Ax) to (Ay); Increment Ax and Ay	
DBcc	W	Dn,address <sup>2</sup>	-----	-	-	-	-	-	-	-	-	-	-	-	-	- 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	---**0	B	-	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	---**0	B	-	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	B	-	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	-----	-	-	-	-	-	-	-	-	-	-	-	-	$\#n \text{ XOR } CCR \rightarrow CCR$	Logical exclusive OR #n to CCR	
EORI	W	#n,SR	-----	-	-	-	-	-	-	-	-	-	-	-	-	$\#n \text{ XOR } SR \rightarrow SR$	Logical exclusive OR #n to SR (Privileged)	
EXB	L	Rx,Ry	-----	B	B	-	-	-	-	-	-	-	-	-	-	register $\leftrightarrow$ register	Exchange registers (32-bit only)	
EXT	WL	Dn	---*00	d	-	-	-	-	-	-	-	-	-	-	-	$Dn.8 \rightarrow Dn.W \mid Dn.W \rightarrow Dn.L$	Sign extend (changes .8 to .W or .L to .L)	
ILLEGAL			-----	-	-	-	-	-	-	-	-	-	-	-	-	PC $\rightarrow$ -(SSP); SR $\rightarrow$ -(SSP)	Generate Illegal Instruction exception	
JMP	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	-	PC $\rightarrow$ -(SP); $\uparrow d \rightarrow PC$	push PC, jump to subroutine at address d	
LEA	L	s,An	-----	-	B	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*	B	-	-	-	-	-	-	-	-	-	-	-		Logical shift Dy, Dx bits left/right	
LSR	W	d	-----	d	-	d	d	d	d	d	d	d	-	-	s		Logical shift Dy, #n bits L/R (#n: 1 to 8)	
MOVE	BWL	s,d	---*00	B	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	$s \rightarrow CCR$	Move source to Condition Code Register	
MOVE	W	s,SR	-----	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	-	-	-	$SR \rightarrow d$	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			

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 <sup>4</sup>	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 <sup>4</sup>	WL	Rn-Rnd s,Rn-Rn	-----	-	-	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+2,An)...(i+4,An) (iAn) → Dn...(i+2,An)...(i+4,An)	Move Dn to/from alternate memory bytes (Access only even or odd addresses)
MOVED <sup>4</sup>	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	*0*0*	d	-	d	d	d	d	d	d	d	-	-	-	0 - d <sub>0</sub> - 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
NDP			-----	-	-	-	-	-	-	-	-	-	-	-	-	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 <sup>4</sup>	BWL	s,Dn Dn,d	---*00	e	-	s	s	s	s	s	s	s	s	s	s	s OR Dn → Dn Dn OR d → d	Logical OR
ORI <sup>4</sup>	BWL	#n,d	---*00	d	-	d	d	d	d	d	d	d	-	-	-	#n OR d → d	Logical OR #n to destination (ORI is used when source is #n)
ORI <sup>4</sup>	B	#n,CCR	---*00	-	-	-	-	-	-	-	-	-	-	-	s	#n OR CCR → CCR	Logical OR #n to CCR
ORI <sup>4</sup>	W	#n,SR	---*00	-	-	-	-	-	-	-	-	-	-	-	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 Dx, Dy bits left/right (without X)
ROR	W	#n,Dy	---	-	-	d	d	d	d	d	d	d	-	-	s		Rotate Dy, #n bits left/right (#n: 1 to 8)
ROXL	BWL	Dx,Dy	---*0*	e	-	-	-	-	-	-	-	-	-	-	-		Rotate Dx, Dy bits L/R, X used then updated
ROXR	W	#n,Dy	---	-	-	d	d	d	d	d	d	d	-	-	s		Rotate Dy, #n bits left/right (#n: 1 to 8)
RTE			---	-	-	-	-	-	-	-	-	-	-	-	-	(SP)+ → SR; (SP)+ → PC	Return from exception (Privileged)
RTR			---	-	-	-	-	-	-	-	-	-	-	-	-	(SP)+ → CCR; (SP)+ → PC	Return from exception (Privileged)
RTS			---	-	-	-	-	-	-	-	-	-	-	-	-	(SP)+ → PC	Return from subroutine and restore CCR
SBCD	B	Dy,Dx -(Ay),-(Ax)	*0*0*	e	-	-	-	-	-	-	-	-	-	-	-	Dx - Dy - X → Dx -(Ax) - -(Ay) - X → -(Ax)	Subtract BCD source and eXtend bit from 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.8 = 11111111 else d.8 = 00000000
STOP		#n	---	-	-	-	-	-	-	-	-	-	-	-	-	#n → SR; STOP	Stop processor (Privileged)
SUB <sup>4</sup>	BWL	s,Dn Dn,d	*****	e	s	d <sup>4</sup>	d	d	d	d	d	d	d	s	s	Dn - s → Dn d - Dn → d	Subtract binary (SUBI or SUBQ used when source is #n. Prevent SUBQ with #n.L)
SUBA <sup>4</sup>	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 <sup>4</sup>	BWL	#n,d	*****	d	-	d	d	d	d	d	d	d	-	-	s	d - #n → d	Subtract immediate from destination
SUBQ <sup>4</sup>	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 -(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		

cc	Condition	Test	cc	Condition	Test
T	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

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 Branch sizes: .B or .S -128 to +127 bytes, .W or .L -32768 to +32767 bytes  
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 AS)  
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