ALGO
QCM

1. **Un sous-graphe G' de G=<S,A> est défini par ?**
   (a) <S,A'> avec A' $\subseteq$ A
   (b) <S',A> avec S' $\subseteq$ S
   (c) <A,S>

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→y ou un arc y→x
   (b) il existe un arc x→y et un arc y→x
   (c) il existe un chemin x⇝y ou un chemin y⇝x
   (d) il existe un chemin x⇝y et un chemin y⇝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→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

1

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→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

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

2

QCM N°9

lundi 30 janvier 2017

Question 11

Soit f continue sur $[a, b[$ où $-\infty < a < b \leqslant +\infty$.

On dit que $\displaystyle\int_a^b f(t)\,\mathrm{d}t$ converge absolument si

a. $\displaystyle\int_a^b f(t)\,\mathrm{d}t$ converge vers un réel positif

b. $\displaystyle\left|\int_a^b f(t)\,\mathrm{d}t\right| = \int_a^b |f(t)|\,\mathrm{d}t$

c. $\displaystyle\int_a^b |f(t)|\,\mathrm{d}t$ converge

d. rien de ce qui précède

Question 12

Soit f positive continue et décroissante sur $\mathbb{R}_+$ telle que la série $\sum f(n)$ diverge. Alors $\displaystyle\int_0^{+\infty} f(t)\mathrm{d}t$ diverge.

a. vrai

b. faux

Question 13

Via le changement de variable $u = t^2$, $\displaystyle\int_1^{+\infty} \frac{\cos(t^2)}{t}\,\mathrm{d}t$ est égale à

a. $\displaystyle\int_1^{+\infty} \frac{\cos(u)}{\sqrt{u}}\,\mathrm{d}u$

b. $\displaystyle\int_1^{+\infty} \frac{\cos(u)}{u}\,\mathrm{d}u$

c. $\displaystyle\int_1^{+\infty} \frac{\cos(u)}{2u}\,\mathrm{d}u$

d. rien de ce qui précède

3

Question 14

Soit f continue et positive sur $[1, +\infty[$ telle que $t^2 f(t) \to 0$ quand $t \to +\infty$. Alors

a. $\displaystyle\int_1^{+\infty} f(t)\mathrm{d}t$ converge

b. $\displaystyle\int_1^{+\infty} f(t)\mathrm{d}t$ diverge

c. on ne peut rien dire sur la nature de $\displaystyle\int_1^{+\infty} f(t)\mathrm{d}t$

Question 15

a. $\forall \alpha > 1$, $\displaystyle\int_1^{+\infty} \frac{\mathrm{d}t}{t^\alpha}$ converge

b. $\forall \alpha < 1$, $\displaystyle\int_1^{+\infty} \frac{\mathrm{d}t}{t^\alpha}$ converge

c. $\forall \alpha \in \mathbb{R}$, $\displaystyle\int_1^{+\infty} \frac{\mathrm{d}t}{t^\alpha}$ diverge

d. $\forall \alpha \in \mathbb{R}$, $\displaystyle\int_1^{+\infty} \frac{\mathrm{d}t}{t^\alpha}$ converge

e. rien de ce qui précède

Question 16

a. $\displaystyle\int_0^{+\infty} e^{-t}\,\mathrm{d}t$ diverge

b. $\displaystyle\int_0^{+\infty} e^{-t}\,\mathrm{d}t$ converge et est égale à -1

c. $\displaystyle\int_0^{+\infty} e^{-t}\,\mathrm{d}t$ converge et est égale à 1

d. rien de ce qui précède

4

Question 17

a. $\displaystyle\int_{-\infty}^{+\infty} t\,\mathrm{d}t = 0$

b. $\displaystyle\int_{-\infty}^{+\infty} t\,\mathrm{d}t$ diverge

c. $\displaystyle\int_{-\infty}^{+\infty} t\,\mathrm{d}t$ converge car $\displaystyle\int_{-x}^{x} t\,\mathrm{d}t$ admet une limite quand x tend vers $+\infty$.

d. rien de ce qui précède

Question 18

Soit f continue sur $]a,b[$ où $-\infty \leqslant a < b \leqslant +\infty$. Supposons qu'il existe $c \in]a,b[$ tel que $\displaystyle\int_c^b f(t)\,\mathrm{d}t$ diverge. Alors on peut directement en conclure que $\displaystyle\int_a^b f(t)\,\mathrm{d}t$ diverge.

a. vrai

b. faux

Question 19

Soient $A \in \mathscr{M}_2(\mathbb{R})$, λ et μ deux valeurs propres réelles distinctes de A. Alors

a. $\mathbb{R}^2 = E_\lambda \oplus E_\mu$

b. E_λ et E_μ sont en somme directe

c. A est diagonalisable

d. rien ce de qui précède

Question 20

Soit $A = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 2 & 0 \\ 3 & 2 & 2 \end{pmatrix}$. Alors

a. -2 est une valeur propre de A

b. 1 est une valeur propre de A

c. 2 est une valeur propre de A

d. -1 est une valeur propre de A

e. rien de ce qui précède

5

QCM Azar Chap13 (Adjec clause ex 38, 40, 43pp288-290)

Read the following sentences, paying attention to their punctuation. Choose the letter of the sentence that gives the correct meaning of the given sentence.

21. The students, who attend class five hours per day, have become quite proficient in their new language.
a. All of the students attend class five hours per day.
b. Some of the students attend class five hours per day.
c. None of the students attend class five hours per day.

22. The orchestra conductor signaled the violinists who were to begin playing.
a. All of the violinists were to begin playing.
b. Some of the violinists were to begin playing.

23. Trees which lose their leaves in winter are called deciduous trees.
a. All trees lose their leaves in winter.
b. Some trees lose their leaves in winter.
c. It is not possible to know which trees lose their leaves in winter.

In 24-26, add commas where necessary.

24. We enjoyed the view from the hotel we stayed in last August.
a. We enjoyed the view, from the hotel we stayed in last August.
b. We enjoyed the view, from the hotel, we stayed in last August.
c. We enjoyed the view, from the hotel we stayed in, last August.
d. No commas.

25. We had to use the toilet, so we walked to the nearest restaurant. The waiter who received us listened sympathetically to our request.
a. No commas.
b. The waiter, who received us listened sympathetically to our request.
c. The waiter who received us, listened sympathetically to our request.
d. The waiter, who received us, listened sympathetically to our request.

26. One of the most common devices found in offices today is the computer which has become indispensable for almost all office work.
a. One of the most common devices, found in offices today is, the computer which has become indispensable for almost all office work.
b. One of the most common devices found in offices today, is the computer which has become indispensable for almost all office work.
c. One of the most common devices found in offices today is the computer which has become indispensable for almost all office work.
d. One of the most common devices found in offices today is the computer, which has become indispensable for almost all office work.

Change these sentences so that the adjective clause contains an expression of quantity.

27. Most of the information we were given was useless.
a. We were given a lot of information, most of it was useless.
b. We were given a lot of information, most of them was useless.
c. We were given a lot of information, most of which was useless.
d. We were given a lot of information, most of which were useless.

28. John gives half of his salary of 7,000 to charity every month.

a. John earns 7,000 a month, half of that he gives to charity.

b. John earns 7,000 a month, half of which he gives to charity.

c. John earns 7,000 a month, half of whom he gives to charity.

d. John earns 7,000 a month half of that is given to charity.

29. Both of the movies I saw last weekend were terrible.

a. I saw two movies last weekend, both of them were terrible.

b. I saw two movies last weekend, both of which were terrible.

c. I saw two movies last weekend, which both they were terrible.

d. I saw two movies last weekend both of them was terrible.

30. Only one of the reasons Claude gave us for not finishing his project was valid.

a. Claude gave us several reasons for not finishing his project, only one of them was valid.

b. Claude gave us several reasons for not finishing his project but one of whose was valid.

c. Claude gave us several reasons for not finishing his project, only one of which was valid.

d. Claude gave us several reasons for not finishing his project, only several of which was valid.

28. John gives half of his salary of 7,000 to charity every month.

a. John earns 7,000 a month, half of that he gives to charity.

b. John earns 7,000 a month, half of which he gives to charity.

c. John earns 7,000 a month, half of whom he gives to charity.

d. John earns 7,000 a month half of that is given to charity.

29. Both of the movies I saw last weekend were terrible.

a. I saw two movies last weekend, both of them were terrible.

b. I saw two movies last weekend, both of which were terrible.

c. I saw two movies last weekend, which both they were terrible.

d. I saw two movies last weekend both of them was terrible.

30. Only one of the reasons Claude gave us for not finishing his project was valid.

a. Claude gave us several reasons for not finishing his project, only one of them was valid.

b. Claude gave us several reasons for not finishing his project but one of whose was valid.

c. Claude gave us several reasons for not finishing his project, only one of which was valid.

d. Claude gave us several reasons for not finishing his project, only several of which was valid.

31 à 40

O.C. MCQ 2

1. In a free market economy, who owns the factors of production?
A) individuals B) firms C) households D) the government

2. The arena of exchange where firms purchase factors of production is called the
A) product market B) factor market C) open market D) production market

3. In general, a household refers to
A) a house B) a person or a group of people living in the same residence C) a person and his friends D) a person and his neighbourhood

4. A household, in economics, is also
A) factors of production B) involved in specialization C) consumers of goods and services
D) called "the invisible hand"

5. The circular flow model of a mixed economy shows
A) how government interacts with households and firms in the product and factor markets
B) how government plans the economy in both the product and factor markets
C) only how firms and households interact in the product and factor markets
D) how taxes and profits are related in the product and factor markets

6. In return for supplying the factors of production, people receive
A) tax breaks B) Social Security C) workers' compensation D) factor payments

7. Each of the following is an example of physical flow in the Factor Market EXCEPT
A) factory labor B) paycheck C) soy from a farm D) used computers

8. The best example of an expenditure from government to households would be
A) an unemployment check B) a fee for a boater's license C) sales tax D) labor force

9. The best example of a physical flow in the product market is
A) a credit card payment B) farm land C) labor for a factory D) an automobile

10. Households provide firms all of the following EXCEPT
A) land B) expenditures C) capital D) labor

SPE
2016/2017

Q.C.M n°9 de Physique

41- Le vecteur densité de courant électronique $\vec{J}$ vérifie :

a) perpendiculaire au vecteur vitesse des porteurs de charges
b) colinéaire et de même sens que le vecteur vitesse des électrons
c) colinéaire et de sens opposé au vecteur vitesse des électrons
d) de sens opposé au sens du courant I

42- Lorsque le vecteur densité de courant $\vec{J}$ est uniforme alors le courant I qui traverse une section circulaire de rayon R d'un conducteur s'écrit :

a) $I = J.2\pi R^2$
b) $I = J.2\pi \mathrm{R}$
c) $I = J.\pi R^2$

43- Aux bornes d'un conducteur rectiligne de longueur L et de conductivité γ à l'intérieur duquel est établi un champ électrique $\vec{E}$ uniforme, la tension U est donnée par

a) $U = \|\vec{E}\|.L$ b) $U = \frac{\|\vec{E}\|}{L}$ c) $U = \gamma.\|\vec{E}\|.L$

44- Un cylindre de rayon R est traversé par une densité de courant $\vec{J}$ variable d'expression $J(r) = J_o.\frac{r}{r_c}.\exp\left(-\frac{r}{r_c}\right)$ où J_o et r_c sont des constantes. Le courant total I traversant le conducteur est

a) $I = J_o.(R - r_c)$ b) $I = J_o.\frac{R}{r_c}$ c) $I = 2\pi.J_o r_c^2(1 - \exp(-\frac{\mathrm{R}}{\mathrm{r_c}}))$

45- La loi d'Ohm s'écrit pour un conducteur de résistance R parcouru par un courant I et avec une tension U

a) $U = R.I^2$ b) $U = R.I$ c) $I = R.U$

46- Une spire de rayon R traversée par un courant I crée en un point M de son axe (Oz) un champ magnétique $\vec{B}(M)$

a) tangentiel
b) nul
c) radial
d) porté par (Oz)

47- La norme du champ magnétique créé au centre d'une spire de rayon R traversée par un courant I est, sachant que $B(z) = \frac{\mu_0 . I . R^2}{2(z^2 + R^2)^{3/2}}$:

a) $B(0) = \frac{\mu_0 . I}{2R}$

b) $B(0) = \frac{\mu_0 . I}{2R^3}$

c) $B(0) = 0$

48- Le champ magnétique $\vec{B}$ généré par un fil infini d'axe (Oz) parcouru par un courant I est

a) radial
b) suivant $\overrightarrow{u_\theta}$
c) parallèle à $\overrightarrow{u_z}$
d) hélicoïdal d'axe (Oz)

49- Sans faire de calcul la loi de Biot-Savart permet de dire que

a) $\vec{B}$ est colinéaire au courant I mais de sens opposé
b) $\vec{B}$ est orthogonal à I
c) $\vec{B}$ est colinéaire au courant I et de même sens

50- On étudie le champ magnétique créé par un courant parcourant un circuit quelconque. On trouve un plan $\mathcal{P}$ de symétrie. Que peut-on dire ?

a) $\vec{B}$ appartient à $\mathcal{P}$
b) $\vec{B}$ est nul
c) $\vec{B}$ est orthogonal à ce plan

QCM Electronique – InfoS4

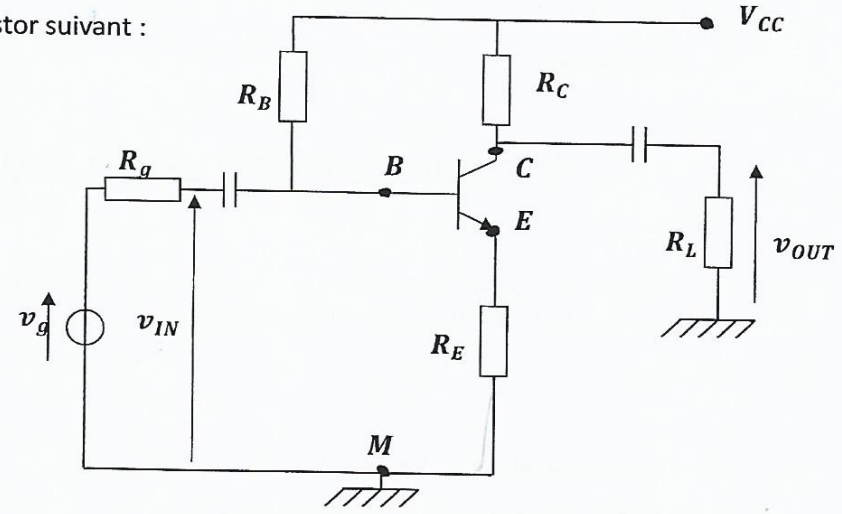
Pensez à bien lire les questions ET les réponses proposées (attention à la numérotation des réponses)

Transistors bipolaires et amplificateur

Q1. En sortie et dans la zone linéaire, le transistor est considéré comme :

a- Une source de tension
b- Une source de courant
c- Une diode

Soit l'amplificateur à transistor suivant :



Q2. A quoi sert la source de tension continue V_{cc}?

a- A rien.
b- A bloquer le transistor.
c- A polariser le transistor dans sa zone linéaire.
d- A saturer le transistor.

Q3. Les condensateurs sont des condensateurs de :

a- découplage
b- liaison.
c- covalence
d- recombination.

Q4. Rôle des condensateurs :

a- Ils ne servent à rien.
b- Ils permettent de couper les composantes variables, car ils sont équivalents à des interrupteurs ouverts en régime variable.
c- Ils bloquent tout type de signal.
d- Ils permettent de couper les composantes continues, car ils sont équivalents à des interrupteurs ouverts en régime continu.

Q5. Pour déterminer le schéma équivalent petits signaux de l'amplificateur :

a- On annule la source de tension variable v_g et on remplace les condensateurs par des fils

b- On annule la source de tension variable v_g et on remplace les condensateurs par des interrupteurs ouverts.

c- On annule la source de tension continue V_{cc} et on remplace les condensateurs par des fils.

d- On annule la source de tension continue V_{cc} et on remplace les condensateurs par des interrupteurs ouverts.

Q6. Dans le schéma équivalent petits signaux, on néglige généralement la résistance de sortie ρ d'un transistor parce qu'elle est :

a- Très petite

b- Très grande

Transistors à effet de champ à jonction

Q7. Un JFET est un composant à haute impédance d'entrée et pouvant être considéré comme une source de courant commandée en courant.

a- Vrai

b- Faux

Q8. Un transistor à effet de champ est un composant à :

a- 3 bornes : la grille, le drain et la source

b- 3 bornes : la base, l'émetteur et le collecteur

c- 2 bornes : l'anode et la cathode

d- 2 bornes : l'émetteur et le récepteur

Q9. Principe de fonctionnement d'un JFET Canal N : Choisir l'affirmation correcte :

a- Le canal, entre le Drain et la Source, constitue un dipôle qui sera conducteur selon la tension v_{GS}.

b- Le canal, entre la Grille et la Source, constitue un dipôle qui sera conducteur selon la tension v_{GS}.

c- Le canal, entre le Drain et la Grille, constitue un dipôle qui sera conducteur selon la tension v_{GS}.

Q10. Pourquoi peut-on considérer le courant de grille comme nul?

a- Parce que l'impédance d'entrée du transistor est nulle

b- Parce que sinon, on ne sait pas faire les calculs

c- Parce que l'impédance d'entrée du transistor est très élevée

d- On ne peut pas considérer le courant de grille comme nul, il est proportionnel à la tension V_{DS}.

QCM 2
Architecture des ordinateurs

Lundi 30 janvier 2017

11. Soient les deux instructions suivantes :

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

Branchement à NEXT si (une ou plusieurs réponses sont possibles) :

A. D1 = $FF0000FF et D2 = $FF0000FF
B. D1 = $FF0000FF et D2 = $00FFFF00
C. D1 = $00FFFF00 et D2 = $00FFFF00
D. D1 = $00FFFF00 et D2 = $FF0000FF

12. Soient les deux instructions suivantes :

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

Branchement à NEXT si (une ou plusieurs réponses sont possibles) :

A. D1 = $FF0000FF et D2 = $FF0000FF
B. D1 = $00FFFF00 et D2 = $FF0000FF
C. D1 = $00FFFF00 et D2 = $00FFFF00
D. D1 = $FF0000FF et D2 = $00FFFF00

13. Soient les deux instructions suivantes :

```
CMP.W D1,D2
BLE   NEXT
```

Branchement à NEXT si (une ou plusieurs réponses sont possibles) :

A. D1 = $FF0000FF et D2 = $FF0000FF
B. D1 = $FF0000FF et D2 = $00FFFF00
C. D1 = $00FFFF00 et D2 = $FF0000FF
D. D1 = $00FFFF00 et D2 = $00FFFF00

14. Soient les deux instructions suivantes :

```
CMP.B D1,D2
BNE   NEXT
```

Branchement à NEXT si (une ou plusieurs réponses sont possibles) :

A. D1 = $FF0000FF et D2 = $FF0000FF
B. D1 = $FF0000FF et D2 = $00FFFF00
C. D1 = $00FFFF00 et D2 = $00FFFF00
D. D1 = $00FFFF00 et D2 = $FF0000FF

15. Soient les deux instructions suivantes :

```
CMP.B D1,D2
BLE   NEXT
```

Branchement à NEXT si (une ou plusieurs réponses sont possibles) :

A. D1 = $FF0000FF et D2 = $00FFFF00
B. D1 = $FF0000FF et D2 = $FF0000FF
C. D1 = $00FFFF00 et D2 = $FF0000FF
D. D1 = $00FFFF00 et D2 = $00FFFF00

14

16. Où se trouvent les *flags* **X**, **N**, **Z**, **V** et **C** ?
    A. Ils se trouvent dans la mémoire RAM.
    B. Ils se trouvent dans le registre USP.
    C. Ils se trouvent dans les 8 bits de poids fort du registre SR.
    D. Ils se trouvent dans les 8 bits de poids faible du registre SR.

17. Où se trouvent les données de la pile du mode utilisateur ?
    A. Elles se trouvent dans le registre PC.
    B. Elles se trouvent dans le registre A7.
    C. Elles se trouvent dans le registre SSP.
    D. Elles se trouvent dans la mémoire RAM.

18. Où se trouvent les données de la pile du mode superviseur ?
    A. Elles se trouvent dans le registre PC.
    B. Elles se trouvent dans le registre A7.
    C. Elles se trouvent dans le registre USP.
    D. Elles se trouvent dans la mémoire RAM.

19. 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.

20. Quel mode a des privilèges limités ?
    A. Le mode utilisateur
    B. Le mode superviseur
    C. Le mode débutant
    D. Le mode noyau

15

16. Où se trouvent les *flags* **X**, **N**, **Z**, **V** et **C** ?
    A. Ils se trouvent dans la mémoire RAM.
    B. Ils se trouvent dans le registre USP.
    C. Ils se trouvent dans les 8 bits de poids fort du registre SR.
    D. Ils se trouvent dans les 8 bits de poids faible du registre SR.

17. Où se trouvent les données de la pile du mode utilisateur ?
    A. Elles se trouvent dans le registre PC.
    B. Elles se trouvent dans le registre A7.
    C. Elles se trouvent dans le registre SSP.
    D. Elles se trouvent dans la mémoire RAM.

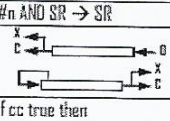
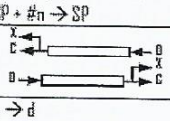
18. Où se trouvent les données de la pile du mode superviseur ?
    A. Elles se trouvent dans le registre PC.
    B. Elles se trouvent dans le registre A7.
    C. Elles se trouvent dans le registre USP.
    D. Elles se trouvent dans la mémoire RAM.

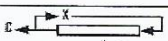
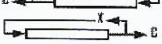
19. Quels modes d'adressage ne spécifient pas d'emplacement mémoire ? (deux réponses)
    A. Mode d'adressage indirect.
    B. Mode d'adressage direct.
    C. Mode d'adressage immédiat.
    D. Mode d'adressage absolu.

20. Quel mode a des privilèges limités ?
    A. Le mode utilisateur
    B. Le mode superviseur
    C. Le mode débutant
    D. Le mode noyau

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 d	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 d	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); Incremeat 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		

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[3]	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[1]	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^{2}	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^{2}	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[a], CC[a]	higher or same	!C	GE	greater or equal	!(N ⊕ V)
LO[a], 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