$egin{array}{c} \mathbf{A}\mathbf{L}\mathbf{G}\mathbf{0} \\ \mathbf{Q}\mathbf{C}\mathbf{M} \end{array}$

1. Le parcours profondeur d'un graphe est par nature?
(a) Récursif
(b) Iteratif
(c) Répétitif
(d) Alternatif
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
o with the state of the foundation of a monthly do?
3. Un graphe orienté de n sommets peut être fortement connexe à partir de?
(a) $n-1$ arcs
(b) n arcs
(c) $n+1$ arcs
4. Deux sommets x et y d'un graphe orienté sont dits adjacents si?
(a) il existe un arc $x \rightarrow y$ ou un arc $y \rightarrow x$
 (b) il existe un arc x→y et un arc y→x
(c) il existe un chemin x→y ou un chemin y→x
(d) il existe un chemin xy et un chemin yx
(d) If existe an chemin xy et an chemin yx
5. Le parcours largeur d'un graphe est par nature?
(a) Récursif
(b) Itératif
(c) Répétitif
(d) Alternatif
(d) Historiasin
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
/_/ Ok

(b) Un graphe partiel(c) Un graphe complet

- 8. L'algorithme de Warshall est utilisable sur?
 - (a) Les graphes orientés statiques
 - (b) Les graphes non orientés statiques
 - (c) Les graphes orientés évolutifs
 - (d) Les graphes non orientés évolutifs
- 9. Supposons que Pref[i] retourne le Numéro d'ordre préfixe de rencontre d'un sommet i. Lors du parcours en profondeur d'un graphe orienté G, les arcs x→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



QCM N°5

Lundi 27 novembre 2023

Question 11

Dans \mathbb{R}^3 , considérons la famille $\mathcal{F} = (u_1 = (1, 1, 0), u_2 = (-1, 1, 1), u_3 = (-1, 1, 0)).$

- a. Cette famille \mathcal{F} est libre
- b. Cette famille \mathcal{F} est liée

Question 12

Dans \mathbb{R}^2 , considérons la base canonique $\mathcal{B} = (e_1 = (1,0), e_2 = (0,1))$, une autre base $\mathcal{B}' = (\varepsilon_1 = (1,-2), \varepsilon_2 = (2,1))$ et un vecteur $u = (x,y) \in \mathbb{R}^2$.

On note $X = \begin{pmatrix} x \\ y \end{pmatrix}$ et $X' = \begin{pmatrix} x' \\ y' \end{pmatrix}$ les colonnes constituées des coordonnées de u dans les bases $\mathcal B$ et $\mathcal B'$.

Alors:

a.
$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

b.
$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

c.
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

d.
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Question 13

Laquelle(Lesquelles) de ces applications est(sont) linéaire(s)?

a.
$$f: \left\{ \begin{array}{ccc} \mathbb{R}[X] & \longrightarrow & \mathbb{R}[X] \\ P & \longmapsto & XP^2 \end{array} \right.$$

b.
$$g: \left\{ \begin{array}{ccc} \mathbb{R}[X] & \longrightarrow & \mathbb{R}[X] \\ P & \longmapsto & X^2P \end{array} \right.$$

c.
$$h: \left\{ \begin{array}{ccc} \mathbb{R}[X] & \longrightarrow & \mathbb{R}[X] \\ P & \longmapsto & XP' + P \end{array} \right.$$

d. Aucune de ces applications n'est linéaire

Question 14

Soit l'application linéaire $f: \left\{ egin{array}{ll} \mathbb{R}[X] & \longrightarrow & \mathbb{R} \\ P & \longmapsto & P(1) \end{array} \right.$. Alors :

a.
$$(1, -2, 1) \in Ker(f)$$

b.
$$(1, -2X, X^2) \in \text{Ker}(f)$$

c.
$$1-2X+X^2 \in \text{Ker}(f)$$

d. Aucun des autres choix

Question 15

Soit l'application linéaire $f:\left\{ egin{array}{ll} \mathbb{R}[X] & \longrightarrow & \mathbb{R} \\ P & \longmapsto & P(1) \end{array}
ight.$ Alors :

- a. f est injective, non surjective
- b. f est surjective, non injective
- c. f n'est ni injective ni surjective
- d. f est bijective

Question 16

Soit $A \in \mathcal{M}_3(\mathbb{R})$. On note C_1 , C_2 et C_3 ses trois colonnes, L_1 , L_2 et L_3 ses trois lignes.

- a. On ne change pas $\det(A)$ si on remplace C_2 par $C_1 + C_2 2C_3$
- b. On ne change pas $\det(A)$ si on remplace C_3 par $C_1+C_2-2C_3$
- c. On ne change pas $\det(A)$ si on remplace L_3 par $L_1-L_2-L_3$
- d. On ne change pas $\det(A)$ si on remplace L_1 par $L_1-L_2-L_3$
- e. Aucun des autres choix

Question 17

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

a.
$$det(A + B) = det(A) + det(B)$$

b.
$$det(\lambda A) = \lambda det(A)$$

c.
$$det(\lambda A) = \lambda^2 det(A)$$

d.
$$det(A \times B) = det(A) \times det(B)$$

e. Aucun des autres choix

Question 18

Soient E un \mathbb{R} -ev, $f \in \mathcal{L}(E)$ et $\lambda \in \mathbb{R}$. Le réel λ est valeur propre de f si et seulement si :

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

Question 19

Soient E un \mathbb{R} -ev, $f \in \mathcal{L}(E)$ et $\lambda \in \mathbb{R}$. On note id l'application identité de E. Le réel λ est valeur propre de f si et seulement si :

- a. $Ker(f \lambda id) = \{0_E\}$
- b $Ker(f \lambda id) \neq \{0_E\}$
- c. $Im(f \lambda id) = \{0_E\}$
- d. $\operatorname{Im}(f \lambda id) \neq \{0_E\}$
- e. Aucun des autres choix

Question 20

Soit $A \in \mathcal{M}_3(\mathbb{R})$. On note I la matrice identité de $\mathcal{M}_3(\mathbb{R})$. Le polynôme caractéristique de A est :

- a. $P_A(X) = \det(XA + I)$
- b. $P_A(X) = \det(XA I)$
- c. $P_A(X) = \det(A + XI)$
- d. $P_A(X) = \det(XA)$
- e. Aucun des autres choix

QCM 7 Azar Chap20 S3 (pg 446 ex44) fall 23

a. had comeb. would come

c. comed. came

Choose the one correct answer for each question. 21. Epita does not have air conditioning, but I wish it _____ some good fans, at least. a. will have b. would have C. had d. has 22. Afshari didn't come to the party on Saturday. Disappointed, you say: I wish _____ a. she had come. b. she came. c. she would have come. d. she will come. 23. It's raining! I wish the sun ____ right now. a. shined b. were shining c. shines d. would be shining 24. I don't know how to play the guitar. I wish I ____ how to play it! a. knew b. will know c. had known d. know 25. Pedro forgot to write down his password. He wishes he _____ to write down his password. a. didn't forget b. had forgotten c. had not forgotten d. would not forget 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. a. had eaten b. had as usual c. would have ate d. ate 27. Declan did not come to Prologin. I wish he _____ to Prologin.

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 (the truck driver) wishes he people around in his car instead of furniture.
a.	had driven
b.	drove
c.	has been driving
d.	drives
20	Maryam can't afford to come on holiday with us but I wish she, because I think we would
	·
	e a great time together.
a.	will come
b.	would come
c.	can come
d.	None of the above.

QCM 7 – OC S3 2023/24 (Week 27 November)

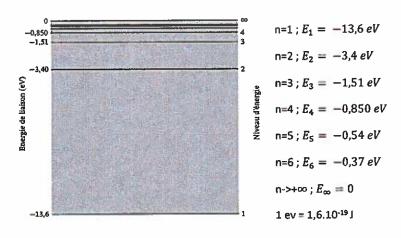
31.	Which of the following is NOT a stage of Oberg's Culture Shock theory?
b) c)	Reverse Shock Honeymoon Adaptation Adjustment
32.	Which of the following according to Oberg are the negative aspects of culture shock? Choose all that apply
b) c)	Stress Anxiety Hair loss Feeling of helplessness
33.	Which is the 3rd stage of Oberg's Culture Shock theory?
b) c)	Reserve Shock Honeymoon Adaptation Adjustment
34.	The repeated stages of culture shock that occurs on returning home is illustrated by which theory?
b) c)	U Curve W Curve Adjustment Adaptation
35.	We are more likely to experience difficulties that are manageable during the Adjustment phase of Oberg's stages of culture shock. True or False?
•	True False
36.	What is the <i>cultural iceberg</i> discussed in the 'Why you'll hate living in Japan' video composed of? <i>Choose all that apply</i> .
b)	Material Aspects Culture Shock Non-Material Aspects Employment options

37. [Video] Mr Eats states the length of time spent in The Honeymoon Phase is the only phase that is similar for everybody. True or False? a) True b) False 38. [Video] According to Mr Eats when you become more comfortable getting help from others and improving your language skills, you are at the _____ phase of culture shock. a) Adaptation b) Delicate c) Adjustment d) Honeymoon 39. [Video] What are some of the problems Mr Eats states that people can encounter in Japan? Choose all that apply a) Impolite locals b) Difficult work culture c) Limited salary options d) Loneliness 40. [Video] What advice is given to help overcome Culture Shock? Choose all that apply a) Accept your culture shock b) Get more comfortable with the language c) Do an activity that makes you happy d) Don't expect too much from the country

QCM Physique - InfoS3 - 27.11

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

Les Q41 à Q43. s'appuient sur le diagramme d'énergie ci-dessous de l'atome d'hydrogène de Bohr.



Q41. La variation d'énergie en jeu pour passer de l'orbite n = 4 à l'orbite n' = 2 est égale à :

- a. -3,4 ev
- b. -0,850 ev
- c. -2,55 ev
- d. 2,55 ev

Q42. Pour arracher un électron à l'atome (ionisation) depuis l'état fondamental vers l'état $n \to +\infty$, la variation d'énergie mise en jeu est :

- a. 13,6 ev
- b. -13,6 ev
- c. 0 ev
- d. 1 ev

Q43. L'on apporte à l'électron dans l'état n = 2 un quanta d'énergie 1 ev. Que se passe-t-il ?

- a. L'électron passe à l'état d'énergie supérieure n = 3.
- b. L'électron passe à l'état d'énergie inférieure n = 1.
- c. If ne se passe rien.
- d. L'électron est arraché à l'atome.

Q44. Le principe d'incertitude d'Heisenberg pour une particule de masse constante m, de vitesse v (donc de quantité de mouvement p = mv), repérée par sa position x, a pour expression (ħ désigne la constante de Planck réduite, et pour une grandeur a, $\Delta \alpha$ désigne son incertitude) :

- b. $\Delta x \Delta v \ge \frac{\tilde{h}}{2}$ c. $m. \Delta x \Delta v \ge \frac{h}{2}$
- d. $m.\Delta x \Delta v = \frac{\hbar}{2}$

Q45. Le principe d'incertitude d'Heisenberg signifie que, pour une particule de masse constante m :

- a. Si l'on connaît sa position avec une grande précision, alors on ne peut pas connaître précisément sa vitesse.
- b. Si l'on connaît sa vitesse avec une grande précision, alors on ne peut pas connaître précisément sa position.
- c. Si l'on connaît sa position avec une grande précision, alors on connaît également sa vitesse avec une grande précision.
- d. Aucune de ces réponses.

Q46. Le principe d'incertitude d'Heisenberg a des effets négligeables sur :

- a. Les objets ayant une faible vitesse
- b. Les objets ayant une grande vitesse
- c. Les objets macroscopiques
- d. Les objets microscopiques

Q47. \mathcal{H} désigne l'opérateur hamiltonien, et E l'énergie d'une particule. L'équation de Schrödinger indépendante du temps, appliquée à la fonction d'onde ψ de cette particule a pour expression :

a.
$$\mathcal{H}\psi = \psi$$

b.
$$\mathcal{H}\psi = E^2\psi$$

c.
$$\mathcal{H}\psi = E\psi^2$$

d.
$$\mathcal{H}\psi = E\psi$$

Q48. ψ désigne la fonction d'onde d'une particule, définie pour $x \in \mathbb{R}$ (cas unidimensionnel). La densité de probabilité de présence de la particule s'obtient en calculant :

a.
$$|\psi(x)|$$

b.
$$|\psi(x)|^2$$

C.
$$\left| \frac{\partial^2 \psi(x)}{\partial x^2} \right|$$

d.
$$\left|\frac{\partial^2 \psi(x)}{\partial x^2}\right|^2$$

Q49. L'intégrale de la densité de probabilité de présence sur l'intervalle] $-\infty$; $+\infty$ [est :

- a. Egale à 0
- b. Egale à 1
- c. Strictement inférieure à 1
- d. Aucune de ces réponses

Q50. On donne la fonction d'onde ψ associée à une particule : $\psi(x) = \psi_0 e^{-\frac{x}{x_0}}$; où x_0 et ψ_0 sont des constantes. On a :

a.
$$\frac{d^2\psi(x)}{dx^2} = \psi_0 x_0^2 e^{-\frac{x}{x_0}}$$

b.
$$\frac{d^2\psi(x)}{dx^2} = \psi_0 x_0 e^{-\frac{x}{x_0}}$$

c.
$$\frac{d^2\psi(x)}{dx^2} = \frac{\psi_0}{x_0^2} e^{-\frac{x}{x_0}}$$

d.
$$\frac{d^2\psi(x)}{dx^2} = -\frac{\psi_0}{x_0^2}e^{-\frac{x}{x_0}}$$

QCM 7

Architecture des ordinateurs

Lundi 27 novembre 2023

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

- 51. Soit l'instruction suivante: MOVEM.L D1-D3/A4/A5,-(A7)
 - Quelle instruction est équivalente?
 - A. MOVEM.L D1/D3/A4-A5,-(A7)
 - B. MOVEM.L D1/D3/A4/A5,-(A7)
 - C. Aucune de ces réponses.
 - D. MOVEM.L A4/A5/D1/D2/D3,-(A7)
- 52. Soient les deux instructions suivantes :

CMP.W D1,D2

BLE NEXT

Branchement à NEXT si:

- A. D1 = \$92181892 et D2 = \$18929218
- B. D1 = \$18929218 et D2 = \$18929218
- C. D1 = \$92181892 et D2 = \$92181892
- D. D1 = \$18929218 et D2 = \$92181892
- 53. Quelle(s) instruction(s) n'est (ne sont) pas possible(s)?
 - A. SUBI.B #2,(A2)
 - B. SUBI.L #42,D0
 - C. SUBI.L D2,D3
 - D. SUBI.L #8,A2
- 54. Quelle(s) instruction(s) n'est (ne sont) pas possible(s)?
 - A. MOVEQ.W #42,D0
 - B. MOVEQ.L D1,D0
 - C. MOVEQ.L #42,D0
 - D. MOVEQ.B #42,D0
- 55. Quelle(s) instruction(s) n'est (ne sont) pas possible(s)?
 - A. MOVEA.B #50,D0
 - B. MOVEA.B #50,A0
 - C. MOVEA.L #50,D0
 - D. MOVEA.L #50,A0

- 56. Trouvez le ou les nombres manquants pour l'addition sur 16 bits suivante afin d'obtenir la bonne combinaison de flags : \$609A + \$? avec N = 1, Z = 0, V = 1, C = 0
 - A. \$1F66
 - B. \$7000
 - C. \$A000
 - D. Aucune de ces réponses
- 57. Trouvez le ou les nombres manquants pour l'addition sur 8 bits suivante afin d'obtenir la bonne combinaison de *flags* : \$71 + \$? avec N = 0, Z = 1, V = 0, C = 0
 - A. \$00
 - B. \$0F
 - C. \$8F
 - D. Aucune de ces réponses
- 58. Quelle est la valeur de D1.L après l'exécution de l'instruction suivante ? SUB.B D0,D1 Valeurs initiales : D0.L = \$00000007, D1.L = \$00000002
 - A. \$0000005
 - B. \$000000FB
 - C. \$FFFFFFB
 - D. Aucune de ces réponses
- 59. Quelle est la valeur de D1.L après l'exécution de l'instruction suivante ? ADD.B D0,D1 Valeurs initiales : D0.L = \$000001F0, D1.L = \$00000111
 - A. \$00000301
 - B. \$00000201
 - C. \$00000101
 - D. Aucune de ces réponses
- 60. Quelle est la valeur de D1.L après l'exécution de l'instruction suivante ? SUB.W D0,D1 Valeurs initiales : D0.L = \$00000007, D1.L = \$00000002
 - A. \$0000FFFB
 - B. \$0000FFF5
 - C. \$000000FB
 - D. Aucune de ces réponses

EAS	y68	K Quic	k Ref	er	еп	ıce	v1.	B	htt	p://ww	v.wo	wgw	ер.со	m/EAS	y68	K.htm Copyrigh	t @ 2004-2007 By: Chuck Kelly
Opcode	Size	Operand	CCR		Effec	tive	Addres							placemen		Operation	Description
	BWL	b,a	XNZVC	On	An	(An)	(An)+	-(An)	(i,An)	(i,An,Rn)	abs.W	abs.L	(i,PC)	(i,PC,Rn)	#п		
ABCD	В	Dy,Dx	*U*U*	е	-	-	-	-	-	-	-	-	-	-	-	$Dy_{10} + Dx_{10} + X \rightarrow Dx_{10}$	Add BCD source and eXtend bit to
		-(Ay)(Ax)		-	-	-	-	е	-	-	-	-	-	- 1	-	$-(Ay)_{10} + -(Ax)_{10} + X \rightarrow -(Ax)_{10}$	destination, BCD result
ADD 4	BWL	s.Dn	****	е	s	2	S	S	S	S	8	S	S	2	S ⁴	s + Dn → Dn	Add binary (ADDI or ADDQ is used when
		On,d	ļ	e	ď	d	d	d	d	d	d	d	-	-	-	Dn + d → d	source is #n. Prevent ADOQ with #n.L)
ADDA ⁴	WL	s.An		S	8	S	8	S	s	S	S	s	2	s	2	s + An → An	Add address (.W sign-extended to .L)
ADDI 4	_	#n,d	****	ď		ď	d	<u>d</u>	d	d	d	d	-	-	_	#n + d → d	Add immediate to destination
A000.4		#n,d	****	ď	d	d	d	d	q	d	ď	d	12	-	_	#n + d → d	Add quick immediate (#n range: 1 to 8)
ADDX		Dy.Dx	****	ē	l-	-	-	-	100		-	-	-	_	_	$0y + 0x + X \rightarrow 0x$	Add source and eXtend bit to destination
ADUX	0"	-(Ay),-(Ax)		-	_	_	-	е		25		_ :	-22	_	_	$-(Ay) + -(Ax) + X \rightarrow -(Ax)$	
AND 4	BWL	s,0n	-**00	e	-	S	S	S	s	S	s	S	s	2	S ⁴	s AND Dn → Dn	Logical AND source to destination
Allo	"""	On.d		e	_	ď	ď	ď	ď	ă	ď	ď	12] -	-	Dn AND d → d	(AND) is used when source is #n)
ANDI ⁴	BWL	#n,d	-**00	ц	-	d	d		d	d	ď	d	-	-	s	#n AND d → d	Logical AND immediate to destination
ANDI 4	B	#n,CCR		-	-	-				- 0	-	-	<u> </u>		s	#n AND CCR → CCR	Logical AND immediate to CCR
ANDI 4	W	#n,58R		ļ-	-	_	-		_			-	10		2	#n AND SR → SR	Logical AND immediate to SR (Privileged)
			****	-		-		-	- Aug	_ -	-	<u> </u>	-	-	_		Arithmetic shift Dy by Dx bits left/right
ASL	BWL	Ox.Oy		9	-	-	-	•	0.000	-	_	-	-	-	-	X ★ 1	Arithmetic shift Dy #n bits L/R (#n: 1 to 8)
ASR	W	#n,Dy		G.	•	<u>.</u>		ď	d	ď	d	d	^	-	2	₽	Arithmetic shift ds I bit left/right (.W only)
D	BM ₃	d address ²		<u> </u>	-	d	9	U	U	U	U	-				if cc true then	Branch conditionally (cc table on back)
Bcc	DW-	adoress	1889 (-	-	-	_	-	•	10.7%	To 1	-	-	- 1	-	1	address -> PC	(8 or 16-bit ± offset to address)
DELTE	D 1	n	*	-	-			1		ı		1				NOT(bit number of d) \rightarrow Z	Set 2 with state of specified bit in d then
BCHG	BL	On,d		e	-	d	l d	d	d	d d	d d	d d	-	529	-		invert the bit in d
0010		#n,d	*=	ď	Ŀ	d	d	d_	d		d	_	-) = //	S	NOT(bit a of d) → bit n of d	·
BCLR	B L	On,d	*	e'	-	d	d	ď	d	ď	d i	ď	•	-	-	NOT(bit number of d) \rightarrow Z	Set Z with state of specified bit in d then
	m7	#n,d		q	-	d	d	d	d	d	d	d	-	-	S	0 → bit number of d	clear the bit in d
BRA	BM ₃	address ²		-	٠	-	-	-	-	-	-	-	-	-	2	address → PC	Branch always (8 or 16-bit ± offset to addr)
BSET	B L	On,d	*	e'	-	d	d	d	d	d	d	d	-	-	-	NOT(bit n of d) \rightarrow Z	Set Z with state of specified bit in d then
		#n,d	<u> </u>	ď	١.	d	d	d	_d	d	d	<u>d</u>	-	-	8	1 → bit n of d	set the bit in d
BSR	BM ₃	address ²		<u> -</u>	-		-	-		20	-	(4)	-	-	-	$PC \rightarrow -(SP)$; address $\rightarrow PC$	Branch to subroutine (8 or 16-bit ± offset)
BTST	18 L	On,d	*	8		d	d	ď	d	d	d	d	ď	d	-	NOT(bit On of d) \rightarrow Z	Set Z with state of specified bit in d
		#n,d		ď	-	d	d	d	<u>d</u>	d	d	q	d	d	8	NOT(bit #n of d) \rightarrow Z	Leave the bit in d unchanged
CHK	W	s,Dn	-*000	e	-	S	2	2	S		S	S	S	S	S	if On<0 or On>s then TRAP	Compare On with O and upper bound (s)
CLR	BWL	d	-0100	<u>d</u>	<u>-</u>	d	d	d	d	d	d	d		-	<u> </u> -	0 → d	Clear destination to zero
CMP 4		s,Dn	_***	9	Σá	S	S	Z	2	S	S	s	S	S	s*	set CCR with On - s	Compare Do to source
CMPA 4	WL	s.An	-***	S	9	S	2	2	S	2	S	2	S	S	S	set CCR with An - s	Compare An to source
CMPI 4	BWL	#n,d	-***	d	-	d	d	d .	В	d	ď	d	-	-	S	set CCR with d - #n	Compare destination to #n
CMPM 4	BWL	(Ay)+,(Ax)+	_***	-	-	-	8	•	33. 5 .7	*	-	-	-	-	-	set CCR with (Ax) - (Ay)	Compare (Ax) to (Ay); Increment Ax and Ay
OBcc	W	Dn,addres ²		-	-	-	-	-	-	-	-	-	-	1.50	Ι-	if cc false then { On-1 → On	Test condition, decrement and branch
																if On ⇔ -1 then addr → PC }	(16-bit ± offset to address)
ZVIO	W	s,Dn	***0	е	-	2	S	2	S	S	2	S	2	s	8	±32bit On / ±16bit s → ±On	Dn= [16-bit remainder, 16-bit quotient]
DIVU	W	s,Dn	-***0	e	-	s	s	2	S	S	S	S	s	s	s	32bit Dn / 16bit s → On	Dn= (16-bit remainder, l6-bit quotient)
EOR 4	BWL	On,d	-**00	E	-5.	d	d	d	d	d	d	d	-	-	s ⁴	On XOR d → d	Logical exclusive OR On to destination
EORI 4		#n,d	-**00	_	-	d	d	ď	d	d	d	ď	-	-	s	#n XOR d → d	Logical exclusive OR #n to destination
EORI 4	В	#n,CCR		† <u>-</u>	-	-	-	-	- 23		-		100	-	S	#n XOR CCR → CCR	Logical exclusive DR #n to CCR
EDRI 4	W	#n,SR		-	-	-	-	-	-		-				S	#n XDR SR → SR	Logical exclusive OR #n to SR (Privileged)
EXG	"	Rx,Ry		е	е	-	-	्	-	<u></u>		1	832	-	 -	register ← → register	Exchange registers (32-bit only)
EXT	WL	Dn	-**00	å	-		2	-27	-	- (5)	-	-	-	100000	+-	On.8 → On.W On.W → On.L	Sign extend (change .8 to .W or .W to .L)
ILLEGAL	WL	UII		<u>u</u>	1.	1050	70	_	<u> </u>		-	-			-	$PC \rightarrow -(SSP); SR \rightarrow -(SSP)$	Generate Illegal Instruction exception
JMP		d		-	-	d	-	-	d	ď	d	- d	d	d	-	1d → PC	Jump to effective address of destination
JSR	\vdash	q		-	+-	d	 	-	d	d	d	d	d	d	1	PC → -(SP); ↑d → PC	push PC, jump to subroutine at address d
				+	+	Ť	ا	_	Ť	_	<u> </u>		 		Ť		
LEA	L)	s.An		-	6	S	-	-	S	2	2	2	S	2	-	↑s → An	Load effective address of s to An
LINK		An,#n		-	-	-	-	-	-	-	-	-	1 -	-	-	$An \rightarrow -(SP)$; $SP \rightarrow An$;	Create local workspace on stack
			1	1			<u> </u>		L		ļ				<u> </u>	$SP + \#n \rightarrow SP$	(negative n to allocate space)
LSL	BWL	Dx.Dy	***0*	е	-	-	•		-	-	-	-	-	-	-	X 0	Logical shift Dy, Dx bits left/right
LSR		#n.Dy		ď	-	-	-	-	-	-		-	-		S	r⇒ X	Lagical shift Dy, #n bits L/R (#n: 1 to 8)
A15000 T	W	d	<u> </u>	-	-	d	d	d	d	d	d	ď	-	-	-		Logical shift d I bit left/right (.W only)
MOVE 4		s,d	-**00	9	s4	8	е	В	е	е	8	9	S	2	S	$z \rightarrow q$	Move data from source to destination
MOVE	W	s,CCR		S	-	2	S	S	S	8	S	S	S	Z	S	s → CCR	Move source to Condition Code Register
MOVE	W	s,SR	====	S	1-	8	S	S	S	8	S	S	s	S	S	$R2 \leftarrow z$	Move source to Status Register (Privileged)
MOVE	W	SR,d		d	-	d	d	d	d	d	ď	d	-	-	-	SR → d	Move Status Register to destination
MOVE	ī	USP,An		-	d	-	-	-	-	-	-	-	-	:×:	-	USP → An	Move User Stack Pointer to An (Privileged)
		An,USP		(%)	2		-	-			-	-	(*)	2.83	-	An → U2P	Move An to User Stack Painter (Privileged)
	BWL		XNZVC	Dn	An	(An)	(An)+	-(An)	(i,An)	(i,An,Rn)	ebs.W	abs.L	(i,PC)	(i,PC,Rn)	#п		,
				_	-	_		<u> </u>			<u> </u>	-			<u> </u>		

Opcode	Pi	Operand	CCR		<u>a</u>		Idduna			1_doction	inn o-	مطفنم	مناحد	placement		Operation	Description
nbcoos	BML						(An)+							(i.PC.Rn)		uperation	DESCRIPTION
HOWELA		s.d	ANZVC	-	_				$\overline{}$	$\overline{}$	_	$\overline{}$	$\overline{}$			- \ \ \ -	Mave source to An (MOVE s.An use MOVEA)
MOVEA*	WL			2	8	S	8	2	Z	S	2	2	S	S		s → An	
MOVEM4		Rn-Rn,d		-	-	ď		4	d	d	d	ď	-	-		Registers \rightarrow d	Move specified registers to/from memory
1453855		s,Rn-Rn		-	-	S	S	-	2	2	2	2	8	S	-	s -> Registers	(.W source is sign-extended to .L far Rn)
MOVEP	WL	Dn.(i,An)		8	-	-	·	-	d	-	-	-	-	•	-	On → (i,An)(i+2,An)(i+4,A.	Move On to/from alternate memory bytes
		(i,An),Dn	++00	q	-	-		•	2	-	-	-	-	-	_	$(i,An) \rightarrow Bn(i+2,An)(i+4,A.$	(Access only even or odd addresses)
MOYEQ*		#n,Dn	-**00	d	-	-	-		-		-		-	-		#n → Dn	Move sign extended 8-bit #n to Dn
MULS		s,Dn	-**00	е	-	\$	8	2	2	S	S	2	8	S			Multiply signed 16-bit; result: signed 32-bit
MULU		s,Dn	-**00	9	-	S	S	S	8	2	2	S	2	2		l6bit s * l6bit On → On	Multiply unsig'd 16-bit; result: unsig'd 32-bit
NBCD		d	*U*U*	d	-	d	d	d	d	d	ď	d	-	٠		D - d ₀ - X → d	Negate BCD with eXtend, BCD result
NEG		d	****	d	-	d	d	ď	d	d	d	d	-	-	-	0-d → d	Negate destination (2's complement)
NEGX	BWL	d	****	d	-	d	ď	ď	đ	d j	d i	ď	•	-	-	O - q - X → q	Negate destination with eXtend
NOP				-	-	-	-	-	-		-	-	-	-	-	None	No operation occurs
NDT	BWL	d	-**00	ď	-	d	d	d	d	d	d	đ	-	-	-	$NOT(q) \rightarrow q$	Logical NOT destination (f's complement)
OR 4	BWL	s,Dn	-**00	В	-	2	S	S	2	S	2	2	2	S	s	s OR On \rightarrow On	Logical OR
		Dn,d		В	-	q	d	d	d	d	d	d	-	-	-	On OR d → d	(ORI is used when source is #n)
ORI 4	BWL	#n.d	-**00	d	-	d	d	d	d	d	d	d	-	-	\$	#n OR d → d	Logical DR #n to destination
DRI 4		#n,CCR	88888	-	-1		-	-	-	- 1	-	-	•	-	S	#n OR CCR → CCR	Logical OR #n to CCR
ORI 4		#n,SR		-	-		-	-	-	-	-	_	-		s	#n OR SR → SR	Logical DR #n to SR (Privileged)
PEA	-	S		-	-	S	-	-	8	s	8	s	2	8	-	↑s → -(SP)	Push effective address of a unto stack
RESET				_		-	-		-	-	-	_	-		-	Assert RESET Line	Issue a hardware RESET (Privileged)
ROL	RWI	Ox.Dy	-**0*	е			-	_	<u> </u>	_	-	_	-	-	-		Rotate Dy, Dx bits left/right (without X)
ROR	UNL	#n,0y	_	ď	_	_	_	_		_	_	_	_		8	[4	Rotate Dy, #n bits left/right (#ń: 1 to 8)
NUN	W	4 1		_		d	d	d	d	d	d	d		_	-		Rotate d 1-bit left/right (.W only)
ROXL		Ox,Dy	***0*	9		-		-	-	-	-	-	-	-	-	- X	Rotate Dy. Dx bits L/R, X used then updated
ROXR	DIIL	#n.By		ď			.		_ '	_		-	_		2	[4]	Rotate Dy, #n bits left/right (#n: 1 to 8)
KUMK	W	q 411203				d	d	đ	d	d	d	đ	_	_			Rotate destination I-bit left/right (.W only)
RTE	111	U	88888	_		<u>u</u>	-	•	-	-		-	-	-	-	(SP)+ → SR; (SP)+ → PC	Return from exception (Privileged)
RTR			##	-		_	-	_	_	<u> </u>	-		_	-	-	$(SP)+ \rightarrow CCR, (SP)+ \rightarrow PC$	Return from subroutine and restore CCR
RTS				-	-	-	-		_		-			_	-	(SP)+ → PC	Return from subroutine
SBCD	8	Dy,Ox	*U*U*	8				-	-	_	-		_	-	-	$0x_m - Dy_m - X \rightarrow Dx_m$	Subtract BCD source and eXtend bit from
วอกก	ū	-(Ay),-(Ax)		2	<u>-</u>	_		e		_	i	_				$-(Ax)_{10}(Ay)_{10} - X \rightarrow -(Ax)_{10}$	destination, BCD result
C	В	d		ď		d	d	ď	d	d	ď	d	_	 -	+-	If cc is true then I's \rightarrow d	If cc true then d.B = 11111111
Sec	Ð	0		u	-	u	"	u	٠	"	"	u	_	-	ľ	else D's \rightarrow d	else d.B = 00000000
0700	-	R			\vdash								-		H	#n → SR: STOP	Move #n to SR, stop processor (Privileged)
STOP	5000	#n	****	-	-	-	-	-	-	-	-	-	-	-	-	Dn - s → Dn	Subtract binary (SUB) or SUBO used when
SUB 4	BWL	s.On		e	S d ⁴	2	S	2	3	2	2	8	8	S	s ⁴	q - Du → q nu - s → nu	source is #n. Prevent SUBQ with #n.1)
		On,d		9		d	d	<u>d</u>	d	d	ď	<u>d</u>	-	-	-		Subtract address (.W sign-extended to .1)
SUBA 4	WL	s,An		S	E	S	S	2	8	8	2	S	8	2		An - s → An	
SUBI 4		#n,d	****	d	-	ď	d	ď	d	ď	d	4	-	-		d - #n → d	Subtract immediate from destination
SNBO 4		#n,d	****	d	ď	d	ď	d	d	d	d	d	-	-	8	d - #n → d	Subtract quick immediate (#n range: 1 to 8)
SNBX	BWL	Dy.Dx	****	6	-	-	-	-	-	-	- 1	-	-	-	-	$Dx - By - X \rightarrow Dx$	Subtract source and extend bit from
	<u> </u>	-(Ay),-(Ax)		-	<u> </u>	-	-	e	<u> </u>	-	-	-			-	$-(Ax)(Ay) - X \rightarrow -(Ax)$	destination
SWAP	W	On	-**00		-	•	-		-	-	<u> </u>	-	-	<u> </u>	ļ-	bits[31:16] ← → bits[15:0]	Exchange the 16-bit halves of On
TAS	В	d	-**00	d	-	d	d	d	d	d	d	d	-	-	Ŀ	10000 7 001111 7 0111 21 2	N and Z set to reflect d, bit7 of d set to 1
TRAP		#n		-	11:-	-	-	-	-	-	-	-	-	-	8	$PC \rightarrow -(SSP); SR \rightarrow -(SSP);$	Push PC and SR, PC set by vector table #n
										L						(vector table entry) \rightarrow PC	(#n range: 0 to 15)
TRAPV	\Box			-	-	-	-			-		_	-			If V then TRAP #7	If overflow, execute an Overflow TRAP
121	BWL	d	-**00	В	-	d	d	d	d	d	d	d			-	test d → CCR	N and Z set to reflect destination
UNLK		An		-	d	-	-	-	-	-	-	-	-	-	-	$An \rightarrow SP; (SP)+ \rightarrow An$	Remove local workspace from stack
	BWL	s.d	XNZVC	On	Åπ	(An)	(An)+	-(An)	(i,An)	(i,An,Rn)	abs.W	abs.L	(i,PC)	(i,PC,Rn)	#n		
				-	•						•				-	• • • • • • • • • • • • • • • • • • • •	100

Condition Tests (+ DR, I NOT, @ XDR; "Unsigned, "Alternate cc.)												
CC	Condition	Test	CC	Condition	Test							
T	true	1	VC	overflow clear	īV							
F	false	0	VS	overflow set	V							
Hla	higher than	!(C + Z)	PL	plus	1N							
TZ _n	lower or same	C+Z	ML	minus	N							
HS", CC"	higher or same	!C	GE	greater or equal	!(N ⊕ V)							
LO", CS*	lower than	C	LT	less than	(N ⊕ V)							
NE	nat equal	17	GT	greater than	$\mathbb{I}[(N \oplus V) + Z]$							
ED	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)

On Bata register (8/16/32-bit, n=0-7)

Rn any data or address register

Source, d Destination

Either source or destination

Immediate data, i Displacement #n

BCD Binary Coded Decimal

Effective address

Long only, all others are byte only

2 Assembler calculates offset PC Program Counter (24-bit) SR Status Register (16-bit)

SSP Supervisor Stack Pointer (32-bit) USP User Stack Pointer (32-bit)

SP Active Stack Pointer (same as A7)

CCR Condition Code Register (lower 8-bits of SR)

N negative, Zzero, V overflow, C carry, X extend-

* set according to operation's result, = set directly - not affected, Dicleared, 1 set, U undefined

Assembler automatically uses A. I. Q or M form if possible. Use #n.L to prevent Quick optimization

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

