TEST

Contrôle janvier 2023

Retour

Temps restant 1:59:29

Question 1

Réponse enregistrée

Noté sur 2,00

Marquer la question

In programming, where do we find the branch conditions?

- [x] a. They are found in the conditions of control structures (if, loops...)
- [ ] b. They are found in the system call instructions
- [ ] c. They are found in the preprocessor conditions (#ifdef ...)
- [ ] d. They are found in the arguments of function calls

Page suivante

Contrôle janvier 2023

Retour

Temps restant 1:47:49

Question 14

Pas encore répondu

Noté sur 2,00

Marquer la question

Indicate if the following instruction has the proposed fields in its coding.

LDR < Rt > , [SP { , # <imm8 > }]

Check all the fields contained in the instruction and only them.

Veuillez choisir au moins une réponse.

- [x] Un codop codé sur 4 à 6 bits
- [x] Un immédiat codé sur 8 bits
- [ ] Un label codé sur 8 bits
- [ ] Un immédiat codé sur 3 bits
- [ ] Un registre SP codé sur 2 bits
- [x] Un champ de registre sur 3 bits
- [ ] Un 2e champ de registre sur 3 bits
- [ ] Un champ registre codé sur 4 bits
- [ ] Une condition codée sur 4 bits
- [ ] Un immédiat codé sur 5 bits
- [ ] Un 3e champ de registre codé sur 3 bits

Page précédente

Page suivante

Retour

Temps restant 1:45:25

Question 15

Réponse enregistrée

Noté sur 2,00

Marquer la question

Indicate if the following instruction has the proposed fields in its coding.

ASRS <Rdn > , <Rm>

Check all the fields contained in the instruction and only them.

Veuillez choisir au moins une réponse.

- [ ] A 3rd register field on 3 bits
- [x] A 2nd register field on 3 bits
- [ ] An 8-bit immediate
- [ ] An SP register coded on 2 bits
- [ ] A label coded on 8 bits
- [x] A codop coded on 4 to 6 bits
- [ ] A 4-bit coded register field
- [x] A 3-bit register field
- [ ] An immediate encoded on 3 bits
- [ ] A condition coded on 4 bits
- [x] An immediate coded on 5 bits

Page précédente

Page suivante

Retour

Temps restant 1:44:34

Question 16

Réponse enregistrée

Noté sur 2,00

Marquer la question

Indicate if the following instruction has the proposed fields in its coding.

SUBS <Rd > , < Rn > ,# <imm3>

Check all the fields contained in the instruction and only them.

Veuillez choisir au moins une réponse.

- [x] Un immédiat codé sur 3 bits
- [ ] Un 3e champ de registre codé sur 3 bits
- [x] Un champ de registre sur 3 bits
- [ ] Un immédiat codé sur 7 bits
- [ ] Un champ registre codé sur 4 bits
- [ ] Un registre SP codé sur 2 bits
- [ ] Une condition codée sur 4 bits
- [ ] Un immédiat codé sur 5 bits
- [ ] Un label codé sur 8 bits
- [ ] Un codop codé sur 4 à 6 bits
- [x] Un 2e champ de registre sur 3 bits

Page précédente

Page suivante

Retour

Temps restant 1:43:00

Question 17

Réponse enregistrée

Noté sur 2,00

Marquer la question

Indicate if the following instruction has the proposed fields in its coding.

ADD [SP ,] SP , # <imm7>

Check all the fields contained in the instruction and only them.

Veuillez choisir au moins une réponse.

- [x] Un immédiat codé sur 7 bits
- [ ] Un champ de registre sur 3 bits
- [ ] Un immédiat codé sur 5 bits
- [ ] Un 2e champ de registre sur 3 bits
- [ ] Un immédiat codé sur 3 bits
- [ ] Un champ registre sur 4 bits
- [ ] Un 3e champ de registre sur 3 bits
- [ ] Un label codé sur 8 bits
- [ ] Une condition codée sur 4 bits
- [ ] Un registre SP codé sur 2 bits
- [ ] Un codop codé sur 4 à 6 bits

Page précédente

Page suivante

Contrôle janvier 2023

Retour

Temps restant 1:42:27

Question 18

Pas encore répondu

Noté sur 2,00

Marquer la question

Indicate if the following instruction has the proposed fields in its coding.

B<c > < l a b e l >

Check all the fields contained in the instruction and only them.

Veuillez choisir au moins une réponse.

- [x] Une condition codée sur 4 bits
- [ ] Un champ registre codé sur 4 bits
- [ ] Un registre SP codé sur 2 bits
- [ ] Un 3e champ de registre sur 3 bits
- [ ] Un immédiat codé sur 5 bits
- [ ] Un immédiat codé sur 3 bits
- [x] Un offset vers un label codé sur 8 bits
- [x] Un codop codé sur 4 à 6 bits
- [ ] Un 2e champ de registre sur 3 bits
- [ ] Un champ de registre sur 3 bits

Page précédente

Page suivante

TEST

Contrôle janvier 2023

Retour

Temps restant 1:40:18

Question 19

Pas encore répondu

Noté sur 2,00

Marquer la question

Give the binary encoding in 2's complement on 8 bits of the negative number - 67 (written in decimal base)

Réponse : 10111101

Page précédente

Page suivante

TEST

Contrôle janvier 2023

Retour

Temps restant 1:39:32

Question 20

Pas encore répondu

Noté sur 1,00

Marquer la question

Consider the operation D2 + 57 where the numbers are written in hexadecimal base.

Using the 8-bit 2's complement coding, indicate if this operation generates a carry.

Veuillez choisir une réponse.

- Vrai
- Faux

Page précédente

Page suivante

TEST

Contrôle janvier 2023

Retour

Temps restant 1:38:54

Question 21

Pas encore répondu

Noté sur 2,00

Marquer la question

An arithmetic shift to the right of 3 positions of a signed binary number coded in 2's complement on 8 bits corresponds to ?

Veuillez choisir au moins une réponse.

- [ ] a. a rotation of the binary configuration
- [ ] b. A division by 2
- [ ] c. A division by 2 to the power of 8
- [ ] d. an overflow
- [x] e. A division by 2 to the power of 3
- [ ] f. A multiplication by 2
- [ ] g. a reset to zero
- [ ] h. A multiplication by 2 to the power of 8
- [ ] i. a change of sign
- [ ] j. A multiplication by 2 to the power of 3

TEST

Contrôle janvier 2023

Retour

Temps restant 1:38:32

Question 22

Pas encore répondu

Noté sur 1,00

Marquer la question

Consider the operation -42 - 87 where the numbers are written in decimal base.

Using the 8-bit 2's complement coding, indicate if this operation generates an overflow.

Veuillez choisir une réponse.

- Vrai
- Faux

Page précédente

Page suivante

TEST

Contrôle janvier 2023

Retour

Temps restant 1:59:14

Question 2

Pas encore répondu

Noté sur 2,00

Marquer la question

In the PARM processor, why is it necessary to store the flags? And where are they stored?

- a. So that the flags are transmitted to the next conditional jump instruction. They are stored in the RAM.
- b. So that the flags are transmitted to the next conditional jump instruction. They are stored in the APSR register.
- c. For the flags to be calculated by the controller. They are stored in the APSR register.
- d. So that the flags are calculated by the controller. They are stored in the RAM.

Page précédente

Page suivante

TEST

Contrôle janvier 2023

Retour

Temps restant 1:58:22

Question 3

Pas encore répondu

Noté sur 2,00

Marquer la question

To which notion does the SP register refer in programming?

Several answers are possible.

- [x] a. SP points to the memory stack of local variables
- [ ] b. SP refers to the notion of program counter
- [ ] c. SP refers to the notion of system protection of the memory
- [x] d. SP is the Stack Pointer
- [ ] e. SP is the address of the function being executed

Page précédente

Page suivante

TEST

Contrôle janvier 2023

Retour

Temps restant 1:57:35

Question 4

Réponse enregistrée

Noté sur 2,00

Marquer la question

In the project how many memory stacks are managed by the processor?

Réponse : 1

Page précédente

Page suivante

TEST

Contrôle janvier 2023

Retour

Temps restant 1:57:00

Question 5

Réponse incomplète

Noté sur 5,00

Marquer la question

Reset | Assembly | Read the instruction | Calculation on data | Shift | Store | Design the next instruction | Write instruction | Load

Recall the different stages of the machine execution cycle studied in class.

1. Read the instruction
2. Load
3. Calculation on data
4. Store
5. Design the next instruction

Page précédente

Page suivante

Question 6

Pas encore répondu

Noté sur 7,00

Marquer la question

Temps restant 1:54:46

Complete and fill in the table associating each processor circuit with the associated execution step from the first question. (Replace in the table by the names you defined in the previous question).

	Read the instruction	Load	Calculation on data	Store	Designate the next instruction
Écriture mémoire RAM	○	○	○	◉	○
Program Counter (PC)	○	○	○	○	◉
UAL	○	○	◉	○	○
Lecture du Banc de registres	○	◉	○	○	○
Lecture Mémoire RAM	○	◉	○	○	○
Lecture mémoire ROM	◉	○	○	○	○
Contrôleur	◉	○	○	○	○

Page précédente

Page suivante

Contrôle janvier 2023

Retour

Temps restant 1:53:56

Question 7

Pas encore répondu

Noté sur 4,00

Marquer la question

Indicate what corresponds to the different types of operands of the ARM v7 instructions.

	Register	Constant	Address	Codop	Condition
<Rt>	◉	○	○	○	○
SP	◉	○	○	○	○
<label>	○	◉	○	○	○
<Imm3>	○	◉	○	○	○
<c>	○	○	○	○	◉
<Rn>	◉	○	○	○	○
<Imm8>	○	◉	○	○	○

Page précédente

Page suivante

TEST

Contrôle janvier 2023

Retour

Temps restant 1:51:07

Question 10

Réponse enregistrée

Noté sur 2,00

Marquer la question

Which component of the PARM architecture receives data directly from the RAM?

Veuillez choisir au moins une réponse.

- [ ] a. The ALU
- [ ] b. The decoder
- [x] c. The register bank
- [ ] d. The ROM
- [ ] e. The controller

Page précédente

Page suivante

TEST

Contrôle janvier 2023

Retour

Temps restant 1:50:18

Question 11

Pas encore répondu

Noté sur 3,00

Marquer la question

In the branching instructions, how is the label encoding calculated?

Veuillez choisir au moins une réponse.

- [x] a. The label value is calculated from the difference between the jump address and the current PC value
- [ ] b. The value of the label is directly the jump address.
- [ ] c. The value of the label depends on the configuration of the flags
- [ ] d. The value of the label is equal to PC + 4
- [ ] e. The value of the label is the address of the Program Counter

Page précédente

Page suivante

TEST

Contrôle janvier 2023

Retour

Temps restant 1:50:03

Question 12

Pas encore répondu

Noté sur 2,00

Marquer la question

What is the Stack Pointer?

Several answers are correct.

Veuillez choisir au moins une réponse.

- [x] a. A specific address within the RAM
- [ ] b. A specific address within the ROM
- [x] c. A register of the processor
- [ ] d. The address of a register
- [ ] e. A pointer to function

Page précédente

Page suivante

Question 24

Pas encore répondu

Noté sur 4,00

Marquer la question

Temps restant 1:30:18

Interpret the following assembly code. What calculation does it perform?

```
.global _start
.global main

_start:                 b           main

main:

LDR R0, cnt
LDR R1, cnt
LDR R2, var1
LDR R3, var2

loop :
ADC R1, R0
SUB R3, R3, R2
CMP R3, R2
Bge loop

STR R1, cnt
STR R3, cnt2

// Allocation et contenu des variables

var2:   .word 0x00186A4

var1:   .word 0x000000A

cnt  :   .word 0x0000000

cnt2 :  .word 0x0000000

.end
```

Check all the answers that are true and only those that are true.

Veuillez choisir au moins une réponse.

- [ ] cnt2 = value of var1
- [ ] cnt = Calculating the subtraction of 100 004 by 10
- [ ] cnt = Calculating the subtraction of 10 004 by 10
- [x] cnt2 = remainder of the division
- [x] cnt = Value of dividing 100 004 by 10
- [x] cnt = number of loop iterations performed
- [ ] cnt2 = number of iterations performed
- [ ] cnt = Calculation of the division of 10 004 by 10
- [ ] cnt2 = subtraction of var 2 - var 1

Question 25

Pas encore répondu

Noté sur 2,00

Marquer la question

Temps restant 1:28:00

Convert this instruction to hexadecimal using the coding tables in the ARM documentation provided below.

LDR R1, cnt

It is assumed for this conversion that the assembler has made the following placement in memory.

Symbol	Memory address in decimal	
Main	0	
Loop	8	
Var2	76	
Var1	72	
cnt	68	
rem	64	

Encoding of thumb instructions -16bits

CODOP	Instructions types		Code ALU	ALU operation
00xxxx	Shift, add, sub, move and compare		0000	Logical And
010000	Data processing		0001	Exclusive logical Or
10010	Store immediate		0010	Logical Shift Left
10011	Load immediate		0011	Logical Shift Right
1101	Conditional Branch		0100	Arithmetic Shift Right
10110000	SP instructions		0101	Add with Carry
<C>	**condition codes**		0110	Subtract with Carry
0000	EQ, Z==1 (EQual)		0111	Rotate Right
0001	NE, Z==0 (Not Equal)		1000	Set flags on bitwise AND
0010	CS, C==1 (Carry Set)		1001	Reverse Subtract from 0
0011	CC, C==0 (Carry Clear)		1010	Compare Registers
0100	MI, N==1 (Minus)		1011	Compare Negative
0101	PL, N==0 (PLus)		1100	Logical OR
0110	VS, V==1 (oVerflow Set)		1101	Multiply Two Registers
0111	VC, V==0 (oVerflow Clear)		1110	Bit Clear
1000	HI, C==1 and Z==0, (HIgher than)		1111	Bitwise NOT

Réponse : 10011 001 00010001

Pas encore répondu

Noté sur 2,00

Marquer la question

It is assumed that the PC of the current instruction is 0. We will assume for simplicity that the memory address encoded in the instruction is the value of the jump tag and not the offset from the PC. We will also neglect the increment of 3 normally used by PARM.

Temps restant 1:26:50

BMI loop

It is assumed for this conversion that the assembler has made a memory placement leading to the following current state.

Symbol	memory Address in decimal
Main	0
Loop	12
Var2	128
Var1	132
cnt	136
rem	142

Thumb instructions encoding - 16bits

CODOP	instructions types		Code ALU	ALU Operation
00xxxx	Shift, add, sub, move and compare		0000	Logical And
010000	Data processing		0001	Exclusive logical Or
10010	Store immediate		0010	Logical Shift Left
10011	Load immediate		0011	Logical Shift Right
1101	Conditional Branch		0100	Arithmetic Shift Right
10110000	SP instructions		0101	Add with Carry
<C>	condition codes		0110	Subtract with Carry
0000	EQ, Z==1 (EQual)		0111	Rotate Right
0001	NE, Z==0 (Not Equal)		1000	Set flags on bitwise AND
0010	CS, C==1 (Carry Set)		1001	Reverse Subtract from 0
0011	CC, C==0 (Carry Clear)		1010	Compare Registers
0100	MI, N==1 (Minus)		1011	Compare Negative
0101	PL, N==0 (PLus)		1100	Logical OR
0110	VS, V==1 (oVerflow Set)		1101	Multiply Two Registers
0111	VC, V==0 (oVerflow Clear)		1110	Bit Clear
1000	HI, C==1 and Z==0, (HIgher than)		1111	Bitwise NOT

Réponse : 1101 0100 00001010

Question 27

Pas encore répondu

Noté sur 2,00

Marquer la question

Convert the following instruction to hexadecimal using the coding tables in the ARM documentation provided below.

SBC R5, R0

Temps restant 1:24:38

It is assumed for this conversion that the assembler has made the following placement in memory.

Symbol	Memory address in decimal
Main	0
Loop	8
Var2	76
Var1	72
cnt	68
rem	64

Encoding of Thumb instructions -16bits

CODOP	Instructions types		Code ALU	ALU Operation
00xxxx	Shift, add, sub, move and compare		0000	Logical And
010000	Data processing		0001	Exclusive logical Or
10010	Store immediate		0010	Logical Shift Left
10011	Load immediate		0011	Logical Shift Right
1101	Conditional Branch		0100	Arithmetic Shift Right
10110000	SP instructions		0101	Add with Carry
<C>	**condition codes**		0110	Subtract with Carry
0000	EQ, Z==1 (EQual)		0111	Rotate Right
0001	NE, Z==0 (Not Equal)		1000	Set flags on bitwise AND
0010	CS, C==1 (Carry Set)		1001	Reverse Subtract from 0
0011	CC, C==0 (Carry Clear)		1010	Compare Registers
0100	MI, N==1 (Minus)		1011	Compare Negative
0101	PL, N==0 (PLus)		1100	Logical OR
0110	VS, V==1 (oVerflow Set)		1101	Multiply Two Registers
0111	VC, V==0 (oVerflow Clear)		1110	Bit Clear
1000	HI, C==1 and Z==0, (HIgher than)		1111	Bitwise NOT

Réponse : 010000 0110 000 101