SULIT

KEMENTERIAN PENDIDIKAN TINGGI
JABATAN PENDIDIKAN POLITEKNIK DAN KOLEJ KOMUNITI

BAHAGIAN PEPERIKSAAN DAN PENILAIAN
JABATAN PENDIDIKAN POLITEKNIK DAN KOLEJ KOMUNITI
KEMENTERIAN PENDIDIKAN TINGGI

JABATAN KEJURUTERAAN ELEKTRIK

PEPERIKSAAN AKHIR

SESI I : 2024/2025

DEC30043: MICROPROCESSOR FUNDAMENTALS

TARIKH : 07 DISEMBER 2024
MASA : 8.30 PAGI – 10.30 PAGI (2 JAM)

Kertas soalan ini mengandungi **SEMBILAN (9)** halaman bercetak.
Bahagian A: Subjektif (4 soalan)
Bahagian B: Esei (1 soalan)
Dokumen sokongan yang disertakan : STM32 Reference Manual

JANGAN BUKA KERTAS SOALAN INI SEHINGGA DIARAHKAN

(CLO yang tertera hanya sebagai rujukan)

SECTION A : 80 MARKS

BAHAGIAN A : 80 MARKAH

INSTRUCTION:

This section consists of **FOUR (4)** subjective questions. Answer **ALL** questions.

ARAHAN :

*Bahagian ini mengandungi **EMPAT (4)** soalan subjektif. Jawab **SEMUA** soalan.*

QUESTION 1

SOALAN 1

CLO1 (a) Give **TWO (2)** functions of Central Processing Unit in microprocessor based system.

*Beri **DUA (2)** fungsi Unit Pemprosesan Pusat dalam sistem mikropemproses.*

[4 marks]

[*4 markah*]

CLO1 (b) Stack is a data structure that operates on the Last In, First Out (LIFO) principle, meaning the last data added to the stack is the first one to be removed. ARM processor uses the main system memory for stack memory operations. With the aids of suitable diagram, explain the pop and push instruction for stack memory operation.

Stack merupakan struktur data yang beroperasi berdasarkan prinsip Last In, First Out (LIFO), yang mana data terakhir yang ditambah ke dalam stack adalah data pertama yang dikeluarkan. Pemproses ARM menggunakan memori sistem utama untuk operasi memori stack. Dengan bantuan rajah yang sesuai, terangkan arahan pop dan push bagi operasi memori stack.

[6 marks]

[*6 markah*]

CLO1

(c) Construct an instruction using ARM Assembly Language based on comments given in Table A1(c)/ below.

Bina aturcara bagi Bahasa pengaturcaraan ARM berdasarkan komen yang diberi dalam jadual A1(c) berikut.

Table A1(c)/Jadual A1(c)

Assembly Language	*Comments*
	Use logical instruction AND for R4 and R5, store at R3. *Gunakan arahan logic DAN ke dalam pendaftar R4 dan R5, simpan pada R3.*
	Use logical instruction OR for R3 and R4, store at R5. *Gunakan arahan logik ATAU ke dalam pendaftar R3 dan R4, simpan pada R5.*
	Use logical instruction EOR for R7 and R8, store at R9. *Gunakan arahan logic E-ATAU ke dalam pendaftar R7 dan R8, simpan pada R9.*
	Use logical instruction BIC for R9, store at R8. *Gunakan arahan logic BIC ke dalam pendaftar R9, simpan pada R8.*
	Use logical instruction OR for R4 and R5, store at R6. *Gunakan arahan logik ATAU ke dalam pendaftar R4 dan R5, simpan pada R6.*

[10 marks]

[*10 markah*]

QUESTION 2

SOALAN 2

CLO1 (a) Identify **TWO (2)** characteristics of Cortex-M3 processor.

*Kenalpasti **DUA (2)** ciri bagi pemproses Cortex-M3.*

[4 marks]

[4 *markah*]

CLO1 (b) One of the most basic functions in an ARM processor is data transfer. Explain **THREE (3)** types of data transfer with an appropriate example (instruction) in Table A2(b) below.

*Antara fungsi pemproses ARM adalah memindahkan data. Terangkan **TIGA (3)** jenis operasi pemindahan data dengan contoh (arahan) yang sesuai di dalam Jadual A2(b) dibawah.*

Table A2(b) / Jadual A2(b)

Data Transfer	Instruction

[6 marks]

[*6 markah*]

CLO1

(c) Construct an instruction using ARM Assembly Language based on comment given in Table A2(c)/ below.

Bina aturcara bagi Bahasa pengaturcaraan ARM berdasarkan komen yang diberi dalam jadual A2(c) dibawah

Table A2(c)/Jadual A2(c)

Assembly Language	*Comments*
	Write data 24 in decimal into register R0, flags get updated. *Tulis data 24 dalam perpuluhan ke dalam daftar R0, bendera akan dikemas kini.*
	Write data FFEE in hexadecimal into register R1 and flags not updated. *Tulis data FFEE dalam heksadesimal ke dalam daftar R1 dan bendera tidak dikemas kini.*
	Move data 1010 1110 in binary into R3 and flags not updated. *Pindahkan data 1010 1110 dalam binari ke dalam daftar R3 dan bendera tidak dikemas kini.*
	Write value of 0xFFFFFFF0 (bitwise inverse of 0xF) to the R2 and update flags. *Tulis nilai 0xFFFFFFF0 (pembalik bit 0xF) ke dalam R2 dan kemas kini bendera.*
	Write data 4092 in decimal into register R1, flags not updated. *Tulis data 4092 dalam perpuluhan ke dalam daftar R1, bendera tidak dikemas kini.*

[10 marks]

[*10 markah*]

QUESTION 3

SOALAN 3

CLO1 (a) Define conditional branch instruction and provide **ONE (1)** example to illustrate its usage in ARM assembly language.

*Takrifkan arahan bercabang bersyarat dan berikan **SATU (1)** contoh yang sesuai bagi menerangkan kegunaannya di dalam Bahasa penghimpun ARM.*

[4 marks]

[*4 markah*]

CLO1 (b) Explain shift and rotate operation with suitable assembly instruction.

Terangkan operasi anjak dan putar berserta arahan Bahasa Penghimpun yang sesuai.

[6 marks]

[6 *markah*]

CLO1 (c) Write an instruction using ARM Assembly Language and Arithmetic operation based on operation given in Table A3(c)/ below.

Tulis aturcara bagi Bahasa pengaturcaraan ARM dan operasi arithmetik berdasarkan operasi yang diberi dalam jadual A3(c) berikut.

Table A3(c)/Jadual A3(c)

Operation	Assembly Language	Arithmetic Operation
ADD		
ADD with Carry		
SUBTRACT		
SUBTRACT with borrow		
MULTIPLY		

[10 marks]

[*10 markah*]

QUESTION 4

SOALAN 4

CLO1 (a) Most conditional branches in ARM processors use flags in the APSR to determine whether a branch should be carried out. Elaborate the flags in conditional branches.

Kebanyakan cabang bersyarat dalam pemproses ARM menggunakan bendera dalam APSR untuk menentukan sama ada sesuatu cabang perlu dilaksanakan. Terangkan bendera-bendera dalam cabang bersyarat.

[6 marks]

[*6 markah*]

CLO1 (b) Explain a comparison operation with zero in conditional branches.
Terangkan operasi perbandingan dengan sifar dalam cabang bersyarat.

[6 marks]

[*6 markah*]

CLO1 (c) In ARM Cortex-M microcontrollers, GPIO stands for General-Purpose Input/Output. It represents a versatile interface that allows the microcontroller to interact with the external world by reading signals from external devices or sending signals. Construct suitable instructions for the following set of register in table A4(c).

*Dalam mikropengawal ARM Cortex-M, GPIO bermaksud Input/Output Pelbagai Tujuan. Ia mewakili antara muka yang serba boleh yang membolehkan mikropengawal berinteraksi dengan dunia luar dengan membaca isyarat dari peranti luaran atau menghantar isyarat. Bina arahan yang sesuai untuk set daftar berikut dala, Jadua*l A4(c).

Table A4(c)/Jadual A4(c)

Set Register	*Instruction*
GPIO port mode register	
GPIO port output type register	
GPIO port output speed register	
GPIO port pull-up/pull-down register	

[8 marks]

[*8 markah*]

SECTION B : 20 MARKS

BAHAGIAN B :20 MARKAH

INSTRUCTION:

This section consists of **ONE (1)** essay question. Answer the question.

ARAHAN:

*Bahagian ini mengandungi **SATU (1)** soalan esei. Jawab soalan tersebut.*

CLO1

QUESTION 1

SOALAN 1

You are an employee at the STM32 Nucleo manufacturer company and need to configure GPIO pins for a specific application. You need to configure pin PC13 as an input to read a signal and pin PB3 as an output to control an external buzzer (active low). Additionally, implement a function to activate and deactivate the buzzer connected to PB3 based on the state of the button connected to PC13. If the button is pressed (PC13 is low), the buzzer will turn on. If the button is not pressed (PC13 is high), the buzzer will turn off. Write the C program with schematic diagram for this application.

Anda merupakan pekerja di syarikat pengeluar STM32 Nucleo dan perlu mengkonfigurasi pin GPIO untuk aplikasi tertentu. Anda perlu mengkonfigurasi pin PC13 sebagai input untuk membaca isyarat dan pin PB3 sebagai output untuk mengawal buzzer luaran (aktif rendah). Selain itu, laksanakan satu fungsi untuk mengaktifkan dan menyahaktifkan buzzer yang disambungkan ke PB3 berdasarkan keadaan butang yang disambungkan ke PC13. Jika butang ditekan (PC13 rendah), buzzer akan dihidupkan. Jika butang tidak ditekan (PC13 tinggi), buzzer akan dimatikan. Lakarkan gambarajah skematik.dan tulis aturcara C bagi aplikasi ini

[20 marks]

[*20 markah*]

SOALAN TAMAT

Table 1 (a) : RCC_IOPENR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOPH EN	Res.	Res.	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN
								rw			rw	rw	rw	rw	rw

Table 1(b) : GPIOx_MODER

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 2y+1:2y **MODEy[1:0]:** Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O mode.
00: Input mode
01: General purpose output mode
10: Alternate function mode
11: Analog mode (reset state)

Table 1(c) : GPIOx_OTYPER

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **OTy:** Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O output type.
0: Output push-pull (reset state)
1: Output open-drain

Table 1(d) : GPIOx_OSPEEDR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEED15 [1:0]		OSPEED14 [1:0]		OSPEED13 [1:0]		OSPEED12 [1:0]		OSPEED11 [1:0]		OSPEED10 [1:0]		OSPEED9 [1:0]		OSPEED8 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEED7 [1:0]		OSPEED6 [1:0]		OSPEED5 [1:0]		OSPEED4 [1:0]		OSPEED3 [1:0]		OSPEED2 [1:0]		OSPEED1 [1:0]		OSPEED0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 2y+1:2y **OSPEEDy[1:0]**: Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O output speed.

00: Very low speed

01: Low speed

10: Medium speed

11: High speed

Note: Refer to the device datasheet for the frequency specifications and the power supply and load conditions for each speed.

Table 1(e) : GPIOx_ODR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **ODy:** Port output data bit (y = 0..15)

These bits can be read and written by software.

Note: For atomic bit set/reset, the OD bits can be individually set and/or reset by writing to the GPIOx_BSRR or GPIOx_BRR registers (x = A..E).

Table 2(a) : GPIOx_PUPDR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 2y+1:2y **PUPDy[1:0]:** Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O pull-up or pull-down

00: No pull-up, pull-down

01: Pull-up

10: Pull-down

11: Reserved

Table 2(b) : GPIOx_IDR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **IDy:** Port input data bit (y = 0..15)

These bits are read-only. They contain the input value of the corresponding I/O port.