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 II : 2025/2026

DEC40283 : MICROPROCESSOR

TARIKH : 11 MEI 2026
MASA : 8.30 PAGI - 10.30 PAGI (2 JAM)

Kertas soalan ini mengandungi **LAPAN (8)** halaman bercetak.
Bahagian A: Struktur (3 soalan)
Bahagian B: Esei (2 soalan)
Dokumen sokongan yang disertakan : GPIO REFFERENCE MANUAL

JANGAN BUKA KERTAS SOALAN INI SEHINGGA DIARAHKAN

(CLO yang tertera hanya sebagai rujukan)

TERBUKA

SECTION A: 60 MARKS

BAHAGIAN A: 60 MARKAH

INSTRUCTION:

This section consists of **THREE (3)** structured questions. Answer **ALL** questions.

ARAHAN:

*Bahagian ini mengandungi **TIGA (3)** soalan struktur. Jawab **SEMUA** soalan.*

QUESTION 1

SOALAN 1

CLO1 (a) The Arithmetic Logic Unit (ALU) is often called the "calculator" of the CPU. However, it cannot function independently. Explain how the Control Unit (CU) and Registers support the ALU during a simple addition operation (R1=R2+R3).

Unit Logik Aritmetik (ALU) sering dipanggil "kalkulator" CPU. Walau bagaimanapun, ia tidak boleh berfungsi secara bebas. Terangkan bagaimana Unit Kawalan (CU) dan Daftar menyokong ALU semasa operasi penambahan mudah (R1=R2+R3).

[5 marks]

[*5 markah*]

CLO1 (b) The ARM Cortex-M programmer's model defines how software interacts with the hardware, focusing on registers, operating modes, and memory layout. The Cortex-M core features a set of 32-bit registers in side the processor to perform data processing and control. Explain the function of the registers below:

- General Purpose Register
- Stack Pointer
- Link register
- Program Counter
- Program Status Register

Model pemprogram ARM Cortex-M menerangkan bagaimana perisian berinteraksi dengan perkakasan, dengan tumpuan kepada pendaftar, mod operasi, dan susunan memori. Teras Cortex-M mempunyai satu set daftar 32-bit dalam pemproses untuk menjalankan pemprosesan data dan kawalan. Terangkan fungsi daftar di bawah:

- *General Purpose Register*
- *Stack Pointer*
- *Link register*
- *Program Counter*
- *Program Status Register*

[5 marks]
[*5 markah*]

CLO1

(c) You are designing a system for an ARM Cortex-M processor that includes 128KB of Flash memory and 64 KB of SRAM. Given :

- **Flash memory** (starting at 0x00000000)
- **SRAM** (starting at 0x20000000)

Draw a detailed memory map diagram for this system. (In your solution identify and label the start and end hexadecimal addresses for the both components)

Anda sedang mereka bentuk sistem untuk pemproses ARM Cortex-M yang merangkumi 1M memori Flash dan 128KB SRAM. Sistem ini juga antara muka dengan peranti GPIO yang terletak di dasar kawasan persisian. Dieberikan:

- **Flash memory** (bermula di alamat 0x00000000)
- **SRAM** (bermula di alamat 0x20000000)

Lukis gambarajah peta memori terperinci untuk sistem ini. (Dalam penyelesaian anda kenalpasti dan label alamat heksadesimal mula dan tamat untuk kedua-dua komponen).

[10 marks]
[*10 markah*]

QUESTION 2

SOALAN 2

CLO1 (a) Cortex M processor has major components such as processor core, register, memory model, Memory Protection Unit(MPU) and debug system. Explain the function of any **TWO (2)** components above.

*Pemproses Cortex M mempunyai komponen utama seperti teras pemproses, daftar, model memori, Unit Perlindungan Memori(MPU) dan sistem nyahpepijat. Terangkan fungsi **DUA(2)** komponen di atas.*

[5 marks]

[*5 markah*]

CLO1 (b) Referring to the flowchart in Figure A2(b) below, explain the sequence of ARM assembly instructions required to execute the decision and arithmetic blocks in the flowchart. Include in your explaination which instruction handles the 'YES' path and which handles the 'NO' path).

Rujuk carta alir dalam Rajah A2(b) di bawah, terangkan urutan arahan pemasangan ARM yang diperlukan untuk melaksanakan blok keputusan dan aritmetik dalam carta alir tersebut. (masukan dalam penerangan anda arahan mana yang mengendalikan laluan 'YA' dan arahan mana yang mengendalikan laluan 'TIDAK').

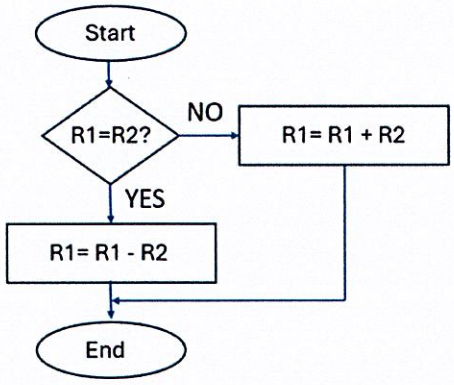

Figure A2(b): Flowchart

Gambarajah A2(b): Carta alir

[5 marks]

[*5 markah*]

CLO1 (c) With the aids of suitable flow chart, write ARM Assembly Language program to inspect the content of R4 and R5 as in Table A2(c) and store the result in register R6.

Condition	*Operation*
R4 is lesser than content of R5	subtract the content of R4 with 0x1
R4 is greater and equal to R5	add the content of R4 with 0x3

Table A2(c)/*Jadual A2(c)*

Dengan bantuan carta alir, tulis satu atucara Bahasa Penghimpunan ARM untuk memeriksa kandungan R4 dan R5 seperti dalam Table A2(C) dan simpan hasilnya dalam daftar R6.

[10 marks]
[*10 markah*]

QUESTION 3
SOALAN 3

CLO1 (a) Based on the AHB-Lite bus protocol used in the Cortex-M3 processor, explain the functions of the I-Code Bus and D-Code Bus.
Berdasarkan protokol bas AHB-Lite yang digunakan dalam pemproses Cortex-M3, terangkan fungsi I-Code Bus dan D-Code Bus.

[4 marks]
[*4 markah*]

CLO1 (b) Determine the result values (1 or 0) of the condition code flags (N, Z, C, and V) in the Program Status Register (xPSR) and register R1 for instruction
SUBS R1, R2 ,R3.(show the calculation method)
Given :R2 and R3 contain 0x0000000A and 0x0000000A, respectively.
Kirakan *Tunjukkan nilai hasil (1 atau 0) bagi bendera kod syarat (N, Z, C, dan V) dalam "Program Status Register" (xPSR) dan Register R3 bagi arahan* ***SUBS R1, R2 ,R3****. (tunjukan jalan kira)*
Diberi : R2 dan R3 masing-masing mengandungi 0x0000000A dan 0x0000000A

[8 marks]
[*8 markah*]

CLO1

(c) In ARM processors, rotate instructions (ROR/RORS) perform circular bit shifting, where bits shifted out from one end re-enter at the opposite end. This differs from standard shift operations where bits are lost. The suffix "S" in an instruction (e.g., RORS) indicates that the instruction updates the APSR (Application Program Status Register) flags. By referring to the following instruction below, analyze the effect of the instruction sequence on register R2 and the APSR flag register. *Dalam pemproses ARM, arahan rotate (ROR/RORS) melaksanakan peralihan bit secara bulatan (circular shift), di mana bit yang keluar dari satu hujung akan masuk semula ke hujung yang bertentangan. Ini berbeza dengan operasi shift biasa di mana bit yang keluar akan hilang. Akhiran "S" dalam sesuatu arahan (contohnya RORS) menunjukkan bahawa arahan tersebut akan mengemaskini APSR (Application Program Status Register). Dengan merujuk kepada arahan di bawah, analisis kesan urutan arahan tersebut terhadap nilai register R2 dan status flag dalam APSR.*

```
Start
LDR R2, = 0XABCDFEED
MOVS R1, #1
RORS R2, R2, R1
END
```


[8 marks]
[*8 markah*]

SECTION B: 40 MARKS

BAHAGIAN B: 40 MARKAH

INSTRUCTION:

This section consists of **TWO (2)** essay question. Answer **ALL** questions.

ARAHAN:

*Bahagian ini mengandungi **DUA(2)** soalan esei. Jawab **SEMUA** soalan.*

QUESTION 1

SOALAN 1

CLO1

An industrial controller stores a data structure at memory address 0x20001000. You are required to load four values from this buffer into registers R2, R3, R4, and R5/R6 respectively.

The data types are:

Status Flag: 8-bit unsigned

Temperature: 16-bit unsigned

Pressure: 32-bit unsigned

Timestamp: 64-bit unsigned

Write an ARM Assembly language program to perform these loads with suitable comment. You must use Post-Indexed Addressing to automatically update the base pointer in R1 after every read operation.

Satu pengawal industri menyimpan struktur data pada alamat memori 0x20001000. Anda dikehendaki memuatkan empat nilai dari penimbal ini ke dalam daftar R2, R3, R4, dan R5/R6 masing-masing.

Jenis data adalah:

Bendera Status: 8-bit tidak bertanda

Suhu: 16-bit tidak bertanda

Tekanan: 32-bit tidak bertanda

Cap Masa: 64-bit tidak bertanda

Tulis satu program bahasa penghimpunan ARM untuk melakukan pemuatan ini. Anda mesti menggunakan Pengalamatan Berindeks Pos untuk mengemas kini penunjuk asas dalam R1 secara automatik selepas setiap operasi bacaan.

[20 marks]
[*20 markah*]

QUESTION 2

SOALAN 2

CLO1 An ARM Cortex-M system is designed with three LEDs connected to PC5, PC6, and PC8 (configured as Push-Pull outputs) and a switch on PC13 (configured without internal Pull-Up or Pull-Down resistors). As an employee at the STM32 Nucleo manufacturer company you need to design a dual-mode logic that differentiates between a 'Running Pattern' (when the switch is active) and a 'Blinking Pattern' (when the switch is inactive). Write the complete C program for this system.

Sistem ARM Cortex-M direka bentuk dengan tiga LED yang disambungkan ke PC5, PC6 dan PC8 (dikonfigurasikan sebagai output Push-Pull) dan suis pada PC13 (dikonfigurasikan tanpa perintang Pull-Up or Pull-Down dalaman). Sebagai pekerja sebagai seorang jurutera di syarikat pengeluar STM32 Nucleo, anda dikehendaki mereka bentuk satu sistem logik dua mod (dual-mode) yang membezakan antara: Mod "Running Pattern" (apabila suis aktif), danMod "Blinking Pattern" (apabila suis tidak aktif). Tuliskan atucara C bagi sistem ini.

[20 marks]
[*20 markah*]

SOALAN TAMAT

RM0377
Reference manual

Ultra-low-power STM32L0x1 advanced Arm®-based 32-bit MCUs

Introduction

This reference manual targets application developers. It provides complete information on how to use the STM32L0x1 microcontroller memory and peripherals.

The STM32L0x1 is a line of microcontrollers with different memory sizes, packages and peripherals.

For ordering information, mechanical and electrical device characteristics please refer to the corresponding datasheets.

For information on the Arm® Cortex®-M0+ core, refer to the *Cortex®-M0+ Technical Reference Manual*.

The STM32L0x1 microcontrollers include state-of-the-art patented technology.

Related documents

- Cortex®-M0+ Technical Reference Manual, available from www.arm.com.
- STM32L0 Series Cortex®-M0+ programming manual (PM0223).
- STM32L0x1 datasheets.
- STM32L0x1 erratasheet.

Bits 3:2 Reserved, must be kept at reset value.

Bit 1 **TIM3RST:** Timer3 reset
Set and cleared by software.
0: No effect
1: Resets timer3

Bit 0 **TIM2RST:** Timer2 reset
Set and cleared by software.
0: No effect
1: Resets timer2

7.3.11 GPIO clock enable register (RCC_IOPENR)

Address: 0x2C

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

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

Bits 31:8 Reserved, must be kept at reset value.

Bit 7 **IOPHEN:** I/O port H clock enable bit
This bit is set and cleared by software.
0: port H clock disabled
1: port H clock enabled

Bits 6:45 Reserved, must be kept at reset value.

Bit 4 **IOPEEN:** I/O port E clock enable bit
This bit is set and cleared by software.
0: port E clock disabled
1: port E clock enabled

Bit 3 **IOPDEN:** I/O port D clock enable bit
This bit is set and cleared by software.
0: port D clock disabled
1: port D clock enabled

8 General-purpose I/Os (GPIO)

8.1 Introduction

Each general-purpose I/O port has four 32-bit configuration registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR and GPIOx_PUPDR), two 32-bit data registers (GPIOx_IDR and GPIOx_ODR) and a 32-bit set/reset register (GPIOx_BSRR). In addition all GPIOs have a 32-bit locking register (GPIOx_LCKR) and two 32-bit alternate function selection registers (GPIOx_AFRH and GPIOx_AFRL).

8.2 GPIO main features

- Output states: push-pull or open drain + pull-up/down
- Output data from output data register (GPIOx_ODR) or peripheral (alternate function output)
- Speed selection for each I/O
- Input states: floating, pull-up/down, analog
- Input data to input data register (GPIOx_IDR) or peripheral (alternate function input)
- Bit set and reset register (GPIOx_ BSRR) for bitwise write access to GPIOx_ODR
- Locking mechanism (GPIOx_LCKR) provided to freeze the I/O port configurations
- Analog function
- Alternate function selection registers
- Fast toggle capable of changing every two clock cycles
- Highly flexible pin multiplexing allows the use of I/O pins as GPIOs or as one of several peripheral functions

8.3 GPIO functional description

Subject to the specific hardware characteristics of each I/O port listed in the datasheet, each port bit of the general-purpose I/O (GPIO) ports can be individually configured by software in several modes:

- Input floating
- Input pull-up
- Input-pull-down
- Analog
- Output open-drain with pull-up or pull-down capability
- Output push-pull with pull-up or pull-down capability
- Alternate function push-pull with pull-up or pull-down capability
- Alternate function open-drain with pull-up or pull-down capability

Each I/O port bit is freely programmable, however the I/O port registers have to be accessed as 32-bit words, half-words or bytes. The purpose of the GPIOx_BSRR register is to allow atomic read/modify accesses to any of the GPIOx_ODR registers. In this way, there is no risk of an IRQ occurring between the read and the modify access.

8.3.15 BOOT0/GPIO pin sharing

On category 1 devices, the BOOT0 pin is shared with a GPIO pin. The BOOT0 pin input level can be read as an input value on the shared GPIO pin. This pin features specific input voltage characteristics (refer to the corresponding datasheet for more details).

8.4 GPIO registers

For a summary of register bits, register address offsets and reset values, refer to *Table 46*.

The peripheral registers can be written in word, half word or byte mode.

8.4.1 GPIO port mode register (GPIOx_MODER) (x =A to E and H)

Address offset:0x00

Reset value: 0xEBFF FCFF for port A

Reset value: 0xFFFF FFFF for the other ports

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 31:0 **MODE[15:0][1:0]:** Port x configuration I/O pin y (y = 15 to 0)
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)

8.4.2 GPIO port output type register (GPIOx_OTYPER) (x = A to E and H)

Address offset: 0x04

Reset value: 0x0000 0000

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

8.4.3 GPIO port output speed register (GPIOx_OSPEEDR) (x = A to E and H)

Address offset: 0x08

Reset value: 0x0C00 0000 (for port A)

Reset value: 0x0000 0000 (for the other ports)

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 31:0 **OSPEED[15:0][1:0]**: Port x configuration I/O pin y (y = 15 to 0)
These bits are written by software to configure the I/O output speed.
00: Low speed
01: Medium speed
10: High speed
11: Very high speed

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

8.4.4 GPIO port pull-up/pull-down register (GPIOx_PUPDR) (x = A to E and H)

Address offset: 0x0C

Reset value: 0x2400 0000 (for port A)

Reset value: 0x0000 0000 (for the other ports)

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 31:0 **PUPD[15:0][1:0]:** Port x configuration I/O pin y (y = 15 to 0)
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

8.4.5 GPIO port input data register (GPIOx_IDR) (x = A to E and H)

Address offset: 0x10

Reset value: 0x0000 XXXX

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 **ID[15:0]:** Port x input data I/O pin y (y = 15 to 0)
These bits are read-only. They contain the input value of the corresponding I/O port.

8.4.6 GPIO port output data register (GPIOx_ODR) (x = A to E and H)

Address offset: 0x14

Reset value: 0x0000 0000

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 **OD[15:0]:** Port output data I/O pin y (y = 15 to 0)
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 register (x = A..E and H).