

## سوالات تئوری تمرین سری دوم سیستم های نهفته

محمد صالح پزند - 400521171

### سوال 1:

پردازنده های عمومی: (General-Purpose Processors)

این پردازنده ها برای اجرای طیف گسترده ای از وظایف طراحی شده اند و معمولاً در رایانه ها، لپ تاپ ها، و سرورها استفاده می شوند.

### ویژگی ها:

- انعطاف پذیری بالا: قابلیت اجرای انواع برنامه ها و وظایف.
- معماری پیچیده: دارای قابلیت های پیشرفته مانند حافظه کش، چند هسته ای بودن، و پردازش موازی.
- سرعت بالا در وظایف کلی: به دلیل فرکانس پردازشی بالا و بهینه سازی های سخت افزاری.

### مزایا:

1. مناسب برای کاربردهای متنوع.
2. پشتیبانی از نرم افزارهای مختلف و سیستم عامل ها.
3. قدرت پردازش بالا برای وظایف پیچیده و چندوظیفه ای.

### معایب:

1. مصرف انرژی بالا: به دلیل طراحی پیچیده و توان پردازشی زیاد.
2. هزینه بیشتر: به دلیل سخت افزار پیشرفته تر.

3. ناکارآمدی در وظایف خاص: در مقایسه با پردازنده‌های وظیفه‌خاص، ممکن است در وظایف تخصصی کارایی بهینه نداشته باشند.

---

### پردازنده‌های وظیفه‌خاص: (Specific-Task Processors)

این پردازنده‌ها برای انجام وظایف خاص یا مجموعه‌ای محدود از وظایف طراحی شده‌اند. نمونه‌هایی از این پردازنده‌ها شامل میکروکنترلرها (Microcontrollers) و DSP (Digital Signal Processors) می‌باشد.

#### ویژگی‌ها:

- طراحی ساده‌تر: متمرکز بر انجام وظایف خاص.
- بهینه‌سازی برای کارایی بالا: در وظایف موردنظر.
- ادغام ویژگی‌های جانبی: مانند حافظه، تایمرها، و ماژول‌های ارتباطی در میکروکنترلرها.

#### مزایا:

1. مصرف انرژی کمتر: به دلیل طراحی بهینه برای وظایف خاص.
2. هزینه پایین‌تر: معمولاً ارزان‌تر از پردازنده‌های عمومی.
3. کارایی بالا در وظایف خاص: به دلیل تمرکز بر نیازهای مشخص.

#### معایب:

1. انعطاف‌پذیری پایین: محدود به وظایف خاص.
2. محدودیت در پردازش: معمولاً قدرت پردازش کمتری در مقایسه با پردازنده‌های عمومی دارند.
3. نیاز به طراحی نرم‌افزار خاص: برای بهره‌گیری کامل از قابلیت‌های سخت‌افزاری.

## سوال 2:

الف) تعامل سیستم‌های نهفته با محیط خارجی و پردازش سیگنال‌های محیطی:

سیستم‌های نهفته (Embedded Systems) برای انجام وظایف خاص طراحی شده‌اند و معمولاً با محیط خارجی تعامل می‌کنند تا داده‌ها را دریافت، پردازش و به خروجی‌های مناسب تبدیل کنند. تعامل این سیستم‌ها با محیط به سه بخش اصلی تقسیم می‌شود:

### 1. دریافت سیگنال‌های محیطی:

- از حسگرها (Sensors) برای دریافت اطلاعات محیطی مانند دما، رطوبت، فشار، نور، صدا، یا حرکت استفاده می‌شود.
- سیگنال‌های آنالوگ حسگرها توسط مبدل‌های آنالوگ به دیجیتال (ADC) به داده‌های دیجیتال قابل پردازش تبدیل می‌شوند.

### 2. پردازش داده‌ها:

- داده‌های دریافتی در پردازنده‌های نهفته (مانند میکروکنترلرها یا میکروپردازنده‌ها) تحلیل و پردازش می‌شوند.
- پردازش شامل اعمال الگوریتم‌های تصمیم‌گیری، فیلتر کردن سیگنال‌ها، یا اجرای وظایف محاسباتی بر اساس داده‌های دریافتی است.

### 3. تولید خروجی:

- پس از پردازش، سیستم می‌تواند با استفاده از عملگرها (Actuators) مانند موتورها، نمایشگرها، یا چراغ‌ها به محیط پاسخ دهد.
  - خروجی‌های دیجیتال ممکن است به مبدل‌های دیجیتال به آنالوگ (DAC) ارسال شوند تا به سیگنال‌های آنالوگ تبدیل شوند.
-

ب) پروتکل‌ها و روش‌های مورد استفاده در تعامل با محیط خارجی:

برای ارتباط مؤثر با حسگرها، عملگرها، و سایر دستگاه‌ها، از روش‌ها و پروتکل‌های زیر استفاده می‌شود:

### 1. روش‌های ارتباط با حسگرها و عملگرها:

- **GPIO (General Purpose Input/Output):** پین‌های چندمنظوره روی میکروکنترلرها برای خواندن سیگنال‌های دیجیتال از حسگرها یا ارسال سیگنال به عملگرها.
- **ADC/DAC:** برای تبدیل سیگنال‌های آنالوگ به دیجیتال و برعکس.
- **PWM (Pulse Width Modulation):** روشی برای کنترل عملگرها مانند موتورهای DC یا LEDها با سیگنال‌های دیجیتال متغیر.

### 2. پروتکل‌های ارتباطی:

- **I<sup>2</sup>C (Inter-Integrated Circuit):**
  - ارتباط دو سیمه برای اتصال چندین حسگر یا دستگاه به یک میکروکنترلر.
  - مناسب برای دستگاه‌های با سرعت پایین تا متوسط.
- **SPI (Serial Peripheral Interface):**
  - ارتباط سریال با سرعت بالا بین میکروکنترلر و دستگاه‌های جانبی.
  - مناسب برای نمایشگرها، حسگرها، و حافظه‌های خارجی.
- **UART (Universal Asynchronous Receiver/Transmitter):**
  - ارتباط سریال برای تبادل داده با دستگاه‌های خارجی یا ماژول‌های ارتباطی.
- **CAN (Controller Area Network):**
  - پروتکل مقاوم برای ارتباط در محیط‌های صنعتی و خودرویی.
  - مناسب برای تبادل داده بین اجزای یک سیستم پیچیده.
- **USB (Universal Serial Bus):**
  - ارتباط استاندارد برای دستگاه‌های پیشرفته‌تر با نیاز به سرعت بالا.
- **ZigBee، Wi-Fi، Bluetooth:** برای ارتباطات بی‌سیم با دستگاه‌ها یا شبکه‌ها.

### 3. پروتکل‌های سطح بالاتر:

- **HTTP/HTTPS:** برای ارتباط با سرورها در سیستم‌های IoT.
- **MQTT (Message Queuing Telemetry Transport):**
  - پروتکل سبک برای تبادل پیام بین دستگاه‌های IoT.
- **Modbus:** پروتکل صنعتی برای ارتباط با PLC ها و سیستم‌های کنترل.

### سوال 3:

**مفهوم Real-Time در پردازنده‌های سیستم‌های نهفته:**

پردازنده‌های **Real-Time** در سیستم‌های نهفته به گونه‌ای طراحی شده‌اند که بتوانند وظایف را در محدوده زمانی مشخص (یا مهلت تعیین شده) اجرا کنند. این بدان معناست که سیستم باید نه تنها نتایج صحیح ارائه دهد، بلکه این نتایج را در زمان مناسب تولید کند.

**ویژگی‌های کلیدی پردازنده‌های Real-Time:**

1. **زمان‌بندی دقیق:**
  - تضمین می‌کند که وظایف در زمان مقرر آغاز و به پایان برسند.
  - از الگوریتم‌های زمان‌بندی خاص مانند **Rate Monotonic Scheduling (RMS)** یا **Earliest Deadline First (EDF)** استفاده می‌شود.
2. **پاسخ‌دهی سریع و تضمین شده:**
  - زمان پاسخ‌دهی به رویدادهای خارجی باید قابل پیش‌بینی باشد.
3. **پیش‌بینی پذیری:**
  - رفتار سیستم در تمامی شرایط قابل پیش‌بینی است، حتی در حالت بار زیاد.
4. **اولویت‌دهی:**
  - وظایف مهم‌تر دارای اولویت بیشتری هستند و ابتدا اجرا می‌شوند.

## انواع سیستم‌های Real-Time:

### 1. سیستم‌های سخت‌افزاری: Real-Time (Hard Real-Time)

- شکست در اجرای وظایف در زمان مقرر ممکن است به نتایج فاجعه‌باری منجر شود.
- مثال: سیستم‌های کنترل پرواز، تجهیزات پزشکی حیاتی، و سیستم‌های ایمنی خودرو.

### 2. سیستم‌های نرم‌افزاری: Real-Time (Soft Real-Time)

- شکست در رعایت مهلت زمانی ممکن است کیفیت را کاهش دهد اما عواقب جدی ایجاد نمی‌کند.
- مثال: پخش ویدئو، بازی‌های رایانه‌ای، یا سیستم‌های مدیریت صف.

## محدودیت‌های پردازنده‌های Real-Time:

### 1. محدودیت‌های سخت‌افزاری:

- قدرت پردازش محدود: ممکن است برای وظایف پیچیده ناکافی باشد.
- منابع حافظه کم: اغلب با محدودیت RAM و ROM مواجه هستند.
- سرعت ثابت کلاک: پردازنده باید با توان سخت‌افزاری خود در زمان‌بندی دقیق کار کند.

### 2. محدودیت‌های نرم‌افزاری:

- زمان‌بندی پیچیده: پیاده‌سازی الگوریتم‌های زمان‌بندی Real-Time دشوار است.
- الگوریتم‌های محدود: فقط الگوریتم‌هایی قابل اجرا هستند که تضمین زمانی بدهند.
- تداخل وظایف: وظایف با اولویت بالا ممکن است وظایف کم‌اولویت را مسدود کنند.

### 3. محدودیت‌های محیطی:

- رویدادهای غیرمنتظره: رویدادهایی مانند افزایش نویز، خطای حسگر، یا قطع برق می‌توانند باعث اختلال شوند.
- نیاز به پایداری بالا: اختلال یا خطا در سیستم می‌تواند باعث از دست رفتن زمان‌بندی و خروجی نادرست شود.

### 4. محدودیت‌های طراحی:

- پیچیدگی طراحی: سیستم باید به طور کامل پیش‌بینی‌پذیر باشد، که طراحی آن را پیچیده‌تر می‌کند.
- هزینه بالا: سیستم‌های Real-Time اغلب نیازمند سخت‌افزار و نرم‌افزار خاص هستند که هزینه تولید را افزایش می‌دهد.

## سوال 4:

### تفاوت‌های بین میکروکنترلر و میکروپروسسور:

میکروکنترلر (Microcontroller) و میکروپروسسور (Microprocessor) هر دو واحدهای پردازشی هستند، اما طراحی و کاربردهای آنها متفاوت است. در ادامه تفاوت‌های کلیدی آنها و موارد کاربردشان شرح داده می‌شود:

#### 1. ساختار و طراحی:

ویژگی	میکروپروسسور (MPU)	میکروکنترلر (MCU)
ترکیب اجزاء	فقط پردازنده است و به حافظه و ماژول‌های خارجی نیاز دارد.	دارای اجزای داخلی مانند پردازنده، حافظه (RAM) و (Flash)، و واحدهای ورودی/خروجی (I/O) است.
پیچیدگی	نیاز به مدارهای جانبی پیچیده‌تر دارد.	طراحی ساده و همه‌چیز در یک تراشه گنجانده شده است.
مصرف انرژی	بالا، به دلیل نیاز به حافظه و اجزای خارجی.	کم، مناسب برای دستگاه‌های کم‌مصرف.
اندازه	بزرگ‌تر به دلیل اجزای خارجی.	کوچک و جمع‌وجور.

#### 2. عملکرد:

ویژگی	میکروپروسسور (MPU)	میکروکنترلر (MCU)
قدرت پردازش	قدرت پردازش بالا برای وظایف پیچیده و چندوظیفه‌ای.	قدرت پردازش کمتری دارد و برای وظایف ساده‌تر طراحی شده است.
سرعت عملکرد	سریع‌تر، مناسب برای برنامه‌های سنگین.	معمولاً پایین‌تر است.
قابلیت چندوظیفه‌ای	برای چندوظیفه‌ای‌ها بهینه شده است.	محدود است.

### 3. هزینه و پیچیدگی طراحی:

ویژگی	میکروپروسسور (MPU)	میکروکنترلر (MCU)
هزینه	گران‌تر، به دلیل نیاز به اجزای خارجی.	ارزان‌تر، به دلیل طراحی یکپارچه.
طراحی سیستم	پیچیده، نیازمند مدارهای جانبی بیشتر است.	ساده، نیاز به قطعات جانبی کمی دارد.

### کاربردها:

#### میکروکنترلر: (Microcontroller)

مناسب برای دستگاه‌های نهفته و سیستم‌هایی که وظایف ساده و مشخصی دارند.

#### • مثال‌های کاربردی:

- دستگاه‌های IoT
- ماشین لباسشویی، مایکروویو، و سایر لوازم خانگی هوشمند.
- سیستم‌های اتوماسیون صنعتی.
- کنترل‌کننده‌های رباتیک.
- دستگاه‌های پزشکی مانند گلوکومتر.
- پروژه‌های آموزشی و نمونه‌سازی سریع (مانند Arduino).

#### میکروپروسسور: (Microprocessor)



مناسب برای سیستم‌هایی که به قدرت پردازشی بالا و قابلیت چندوظیفه‌ای نیاز دارند.

• مثال‌های کاربردی:

- رایانه‌های شخصی و لپ‌تاپ‌ها.
- سرورها و پایگاه‌های داده.
- تلفن‌های هوشمند و تبلت‌ها.
- سیستم‌های پیچیده صنعتی و پردازش تصویر.
- تجهیزات شبکه و ارتباطات.