**گزارش کار آزمایش شماره 8**

اسم و اسم خانوادگی: عبدالقدیر فرتاش شماره دانشجویی: 99243100

اسم و اسم خانوادگی: عقیل فایق شماره دانشجویی: 99243104

سوالات تحلیلی:



قسمت اول) منابع کالک میکروکنترلر STM32F401 می‌تواند شامل موارد زیر باشد:

1. **کور مرکزی (CPU)**:
   * میکروکنترلر STM32F401 از هسته Cortex-M4 استفاده می‌کند. این هسته دارای واحد همگانی (ARM Cortex-M) با عملکرد بالا و قابلیت‌های پردازشی و کنترلی 32 بیتی است.
2. **حافظه**:
   * **حافظه برنامه (Flash)**: STM32F401 دارای حافظه Flash داخلی با ظرفیت متغیر است که به طور معمول بین 128 کیلوبایت تا 512 کیلوبایت است. این حافظه برای ذخیره کد برنامه و داده‌های ثابت استفاده می‌شود.
   * **حافظه رم (RAM)**: میکروکنترلر دارای حافظه رم داخلی با ظرفیت متغیر است که برای ذخیره داده‌های متغیر و پشتیبانی از اجرای برنامه‌ها استفاده می‌شود.
3. **واحد‌های ورودی/خروجی (I/O)**:
   * STM32F401 دارای واحدهای GPIO (ورودی/خروجی عمومی) است که به شما امکان کنترل دستگاه‌ها و ارتباط با سایر قطعات را می‌دهد.
4. **واحدهای ارتباطی**:
   * **UART/USART**: برای ارتباط سریال با سایر دستگاه‌ها.
   * **SPI (Serial Peripheral Interface)**: برای ارتباط با دستگاه‌های جانبی مانند حافظه‌های فلش و حسگرها.
   * **I2C (Inter-Integrated Circuit)**: برای اتصال به دستگاه‌های I2C مانند سنسورها و EEPROMs.
   * **CAN (Controller Area Network)**: برای ارتباط در شبکه‌های کنترل خودروها و سیستم‌های صنعتی.
   * **USB (Universal Serial Bus)**: برای ارتباط با کامپیوترها و دیگر دستگاه‌های USB.
5. **واحدهای تایمر و شمارنده**:
   * برای تولید سیگنال‌های تایمر و شمارش و زمان‌سنجی در برنامه‌های زمان‌بندی و کنترل.
6. **مبدل‌های آنالوگ به دیجیتال (ADC) و دیجیتال به آنالوگ (DAC)**:
   * برای اندازه‌گیری سیگنال‌های آنالوگ و تولید سیگنال‌های آنالوگ برای کنترل خروجی.
7. **واحدهای PWM (پالس-عرض مدوله)**:
   * برای تولید سیگنال‌های PWM برای کنترل سرعت موتورها و سیستم‌های قدرت.

همچنین، ممکن است میکروکنترلر STM32F401 دارای ویژگی‌های دیگری مانند مبدل‌های آنالوگ به دیجیتال با رزولوشن بالا، پشتیبانی از ایستگاه‌های اندازه‌گیری دما و دیگر سنسورها، و قابلیت‌های امنیتی مانند رمزنگاری و حفاظت در برنامه‌های بیشتری نیز می‌تواند داشته باشد.



قسمت دوم)

جدول وکتور اینتراپت یا به اصطلاح "Vector Interrupt Table" در پردازنده‌های ARM Cortex-M یک ساختار داده‌ای است که آدرس‌های وکتورهای اینتراپت مختلف را در خود ذخیره می‌کند. این جدول معمولاً در حافظه‌ای ثابت (مانند Flash) ذخیره می‌شود و زمانی که یک وقفه رخ می‌دهد، پردازنده به طور خودکار به آدرس مربوط به وکتور اینتراپت می‌رود

**عملکرد جدول وکتور اینتراپت:**

1. **تعریف وکتورهای اینتراپت**:
   * برنامه‌نویس، وکتورهای اینتراپت مربوط به وقفه‌های مختلف را در جدول وکتور اینتراپت تعریف می‌کند. هر وکتور معمولاً به یک وقفه خاص مرتبط است.
2. **ذخیره آدرس‌های وکتورها**:
   * آدرس‌های وکتورهای اینتراپت در جدول وکتور اینتراپت ذخیره می‌شوند. این آدرس‌ها معمولاً به آدرس‌های حافظه‌ای از نوع Flash یا RAM اشاره دارند که کد مربوط به پردازش وقفه در آنجا ذخیره شده است.
3. **انتقال به وکتور اینتراپت**:
   * زمانی که یک وقفه رخ می‌دهد، پردازنده به طور خودکار به آدرس مربوط به وکتور اینتراپت مراجعه می‌کند. این آدرس از جدول وکتور اینتراپت بازیابی می‌شود.
4. **پردازش وقفه**:
   * پس از رفتن به آدرس وکتور اینتراپت، کنترل به کد مخصوصی که در آن آدرس ذخیره شده است، منتقل می‌شود. این کد معمولاً یک تابع (معمولاً به عنوان "روتین اینتراپت") است که برای پردازش وقفه مربوطه طراحی شده است.
5. **بازگشت به برنامه اصلی**:
   * پس از انجام عملیات پردازش، کنترل به برنامه اصلی باز می‌گردد و اجرای برنامه اصلی ادامه می‌یابد.

**نکات مهم:**

* جدول وکتور اینتراپت باید در محدوده حافظه‌ای ثابتی (مانند یک قسمت از Flash) قرار گیرد و باید به طور صحیح در زمان بارگذاری برنامه در میکروکنترلر پر شود.
* اختصاص دادن آدرس‌های وکتور اینتراپت به درستی بسیار حیاتی است؛ در غیر این صورت، پردازنده ممکن است به آدرس‌های نادرست مراجعه کند و عملکرد صحیح را تضمین نکند.
* برنامه‌نویس باید اطمینان حاصل کند که کدهای مربوط به پردازش وقفه‌ها به طور صحیح در محل مربوطه در حافظه قرار داده شده و به درستی عمل می‌کنند.

قسمت عملی و پیاده سازی)

خروجی شبه سازی در پروتیوس:



