Card Verifier



**محققین : متین امیرپناه فر - نیما مخملی**

**استاد درس: دکتر حاتم عبدلی**

فهرست  
  
مقدمه - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  
  
  
  
  
  
*منابع* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
***مقدمه***  
  
پروژه **Card Verifier** یک سیستم دیجیتال است که ورودی عددی کارت را دریافت کرده، اعتبارسنجی می‌کند و در صورت معتبر بودن، سیگنال تأیید را فعال می‌کند. این سیستم از چندین ماژول مجزا تشکیل شده و در معماری **Structural** پیاده‌سازی شده است.  
  
  
  
  
  
در این پروژه، معماری به‌صورت **Structural** پیاده‌سازی شده است که شامل سه ماژول اصلی است:

1️⃣ **ماژول اصلی** (CardVerifier)  
2️⃣ **ماژول پردازش داده** (ChunkProcessor)  
3️⃣ **ماژول اعتبارسنجی** (Validator)

### **توضیح دقیق ماژول‌های پروژه Card Verifier**

#### ****ماژول اصلی (CardVerifier)****

این ماژول به‌عنوان ماژول سطح بالا عمل می‌کند که دو کامپوننت ChunkProcessor و Validator را در خود جای داده است. وظایف اصلی آن شامل دریافت ورودی‌های کلاک، ریست، داده‌ی کارت و سیگنال آماده‌بودن داده است. داده‌های ورودی را به پردازشگر چانک ارسال می‌کند و پس از دریافت خروجی از ماژول اعتبارسنج، مقدار نهایی را در سیگنال valid\_out قرار می‌دهد. اگر مقدار خروجی اعتبارسنجی یک باشد، یعنی شماره کارت معتبر است و در غیر این‌صورت نامعتبر خواهد بود. این ماژول همچنین وظیفه برقراری ارتباط بین ماژول‌های ChunkProcessor و Validator را دارد و تمام پردازش‌های سیستم را هماهنگ می‌کند.

#### ****ماژول پردازش داده (ChunkProcessor)****

وظیفه این ماژول دریافت چهار مقدار چهار بیتی از شماره کارت، جمع کردن آنها و ارسال نتیجه به اعتبارسنج است. در هر کلاک، مقدار chunk\_in دریافت شده و به مقدار متغیر chunk\_sum اضافه می‌شود. در همین حال شمارنده chunk\_counter مقدار خود را افزایش می‌دهد تا مشخص شود چند چانک پردازش شده است. پس از پردازش چهار چانک، مقدار chunk\_sum نهایی به ماژول اعتبارسنج ارسال می‌شود. در صورت ریست شدن، مقادیر شمارنده و مجموع داده‌ها صفر می‌شوند تا آماده پردازش یک شماره کارت جدید باشند. این ماژول درواقع داده‌های کارت را به‌صورت مرحله‌ای دریافت و پردازش کرده و نتیجه‌ی محاسبه‌شده را به مرحله بعدی ارسال می‌کند.

#### ****ماژول اعتبارسنجی (Validator)****

این ماژول وظیفه بررسی اعتبار شماره کارت را بر عهده دارد. مقدار chunk\_sum دریافت‌شده از ماژول پردازشگر چانک را بررسی می‌کند که آیا بر عدد ۱۰ بخش‌پذیر است یا خیر. در صورتی که شرط برقرار باشد، خروجی is\_valid مقدار یک خواهد داشت و در غیر این‌صورت مقدار صفر خواهد گرفت. در مرحله بعد، مقدار is\_valid به ماژول اصلی CardVerifier ارسال می‌شود تا نتیجه‌ی اعتبارسنجی کارت مشخص شود. این ماژول از یک شرط ساده ریاضی برای بررسی درستی شماره کارت استفاده می‌کند و منطق آن به‌راحتی قابل گسترش و تغییر است.

#### ****نحوه ارتباط بین ماژول‌ها****

ارتباط بین ماژول‌های مختلف از طریق سیگنال‌های داخلی انجام می‌شود. مقدار chunk\_in از CardVerifier به ChunkProcessor ارسال می‌شود. پس از محاسبه مجموع چانک‌ها در ChunkProcessor، مقدار chunk\_sum به Validator منتقل می‌شود تا بررسی صحت انجام شود. در نهایت مقدار is\_valid از Validator به CardVerifier ارسال شده و در سیگنال valid\_out تنظیم می‌شود. این ساختار ماژولار باعث افزایش خوانایی، قابلیت نگهداری و توسعه‌پذیری کد شده و امکان تغییر یا بهینه‌سازی هر بخش را بدون تأثیر بر بخش‌های دیگر فراهم می‌کند.

### 

## **. ماژول اصلی (CardVerifier)**

این ماژول به‌عنوان **ماژول بالادستی** عمل کرده و دو کامپوننت ChunkProcessor و Validator را در خود قرار داده است. وظایف اصلی آن شامل موارد زیر است:

* دریافت ورودی‌های **کلاک، ریست، داده‌ی کارت، و سیگنال آماده‌بودن داده**
* ارسال داده به **پردازشگر چانک‌ها (ChunkProcessor)** برای جمع‌آوری اطلاعات
* دریافت خروجی chunk\_sum از پردازشگر و ارسال آن به **اعتبارسنج (Validator)**
* در نهایت، تعیین خروجی valid\_out برای مشخص‌کردن معتبر بودن یا نبودن شماره کارت

### **نحوه عملکرد:**

هنگام دریافت data\_ready=1، داده‌ی ۴ بیتی به **پردازشگر چانک** ارسال می‌شود.  
 پردازشگر چانک مقدار عددی کارت را در یک متغیر ذخیره و پردازش می‌کند.  
 پس از پردازش چهار چانک، مقدار **جمع عددی کارت** به **ماژول اعتبارسنجی** ارسال می‌شود.  
اگر مقدار chunk\_sum mod 10 = 0 باشد، کارت معتبر (valid\_out=1) در غیر این‌صورت نامعتبر (valid\_out=0) خواهد بود.

### **ماژول پردازش داده (ChunkProcessor)**

* در هر کلاک، مقدار ۴ بیتی chunk\_in را دریافت می‌کند و مجموع تمام چانک‌ها را ذخیره می‌کند.
* هر ۴ داده‌ی ۴ بیتی را جمع کرده و در سیگنال chunk\_sum ذخیره می‌کند.
* شمارنده chunk\_counter را افزایش می‌دهد تا تعداد چانک‌های پردازش‌شده را بشمارد.
* در پایان ۴ مرحله، مقدار نهایی را به ماژول بعدی ارسال می‌کند.

این ماژول وظیفه دارد که **چهار مقدار ۴ بیتی (چانک) ورودی را دریافت و جمع کند** و در نهایت، مقدار نهایی را به اعتبارسنج ارسال کند.

### **نحوه عملکرد:**

✅ مقدار عددی chunk\_in در هر کلاک **به مقدار chunk\_sum اضافه می‌شود** تا مجموع نهایی محاسبه گردد.  
✅ شمارنده‌ی chunk\_counter تعداد چانک‌های پردازش‌شده را ثبت می‌کند.  
✅ بعد از دریافت چهار چانک، مقدار chunk\_sum به **ماژول اعتبارسنجی** ارسال می‌شود.

### **روند پردازش در فلوچارت:**

دریافت مقدار chunk\_in از CardVerifier  
 افزایش مقدار chunk\_sum با مقدار ورودی  
 افزایش مقدار chunk\_counter  
 پس از پردازش ۴ چانک، مقدار chunk\_sum را به **اعتبارسنج** ارسال می‌کند

## **ماژول اعتبارسنجی (Validator)**

### **عملکرد:**

این ماژول وظیفه دارد **اعتبار شماره کارت را بررسی کند**. منطق بررسی به این صورت است:

* اگر مقدار chunk\_sum mod 10 = 0 باشد، شماره کارت معتبر است (valid\_out=1).
* در غیر این‌صورت، شماره کارت نامعتبر است (valid\_out=0).

منابع   
  
آموزش های ویدیویی