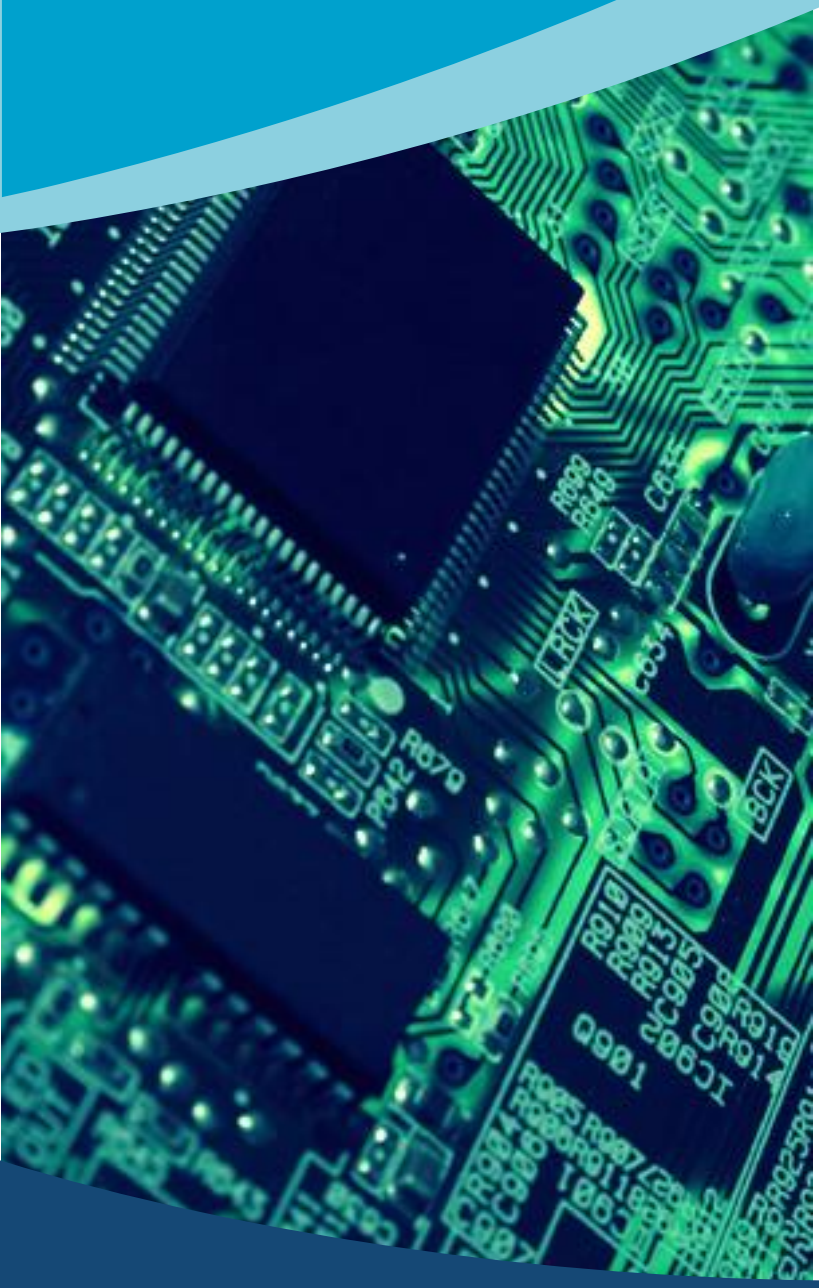
IEEE

DIGITAL SYSTEMS

سیستم های دیجیتال



سیستم های دیجیتال رو راست باشیم یه چیز عجیب غریبه. یه گرایش کاملا بومیه که هیچ جا جز ایران تقریبا پیدا نمیشه و حتی توی ایران هم دانشگاه به دانشگاه با کمی اغراق از زمین تا آسمون فرق داره.

هدفش چیه؟ راستش سخته جواب این سؤال رو داد. بذارید اول یه تصویر ذهنی براتون بسازم!

اگه یه هرم در نظر بگیرید که بالاترین خونه‌ش برنامه نویسای بی خبر از همه چی باشن که فقط سینتکس بلد باشن و پایین ترین خونه‌ش ترانزیستورا باشند، یه جا اون وسط آدمایی هستن که کارشون سیستم عامله و کامپایلره، زیر سیستم عاملی ها و کامپایلری ها معمارای کامپیوترن، زیر معماری کامپیوتر هم قراره مکان فرضی سیس دیجی ها باشه. یعنی سیس دیجی ها مثلا مهندسای سخت افزاری هستن سطح پایین تر فک میکنند و بیشتر با ترانزیستور و گیت سرکار دارن .

البته این تعریف سیس دیج دانشگاه تهرانه و با بقیه جاها فرق داره.

شاید بعضیا سیس دیج رو کنار الکترونیک مقایسه کنن، این کار راستش خیلی درست نیست. اشتراک سیس دیج با الکترونیک خیلی کمه. در واقع اگه قبلنا سخت افزار پلی بین برق و نرم افزار بود، با این ابتکار مثلا سیس دیج شده پل بین برق و سخت افزار(:

اگه عشق ریاضی، عشق آنالوگ، عشق مکانیک و کنترل و قدرت و رباتیک و این چیزا هستید متاسفانه یا خوشبختانه سیس دیج جای شما نیست.

سیس دیج واسه اونایی‌ست که میخوان بشینن پشت کامپیوتر کد بزنن اما نه اپلیکشین اندروید و آیفون، نه بک اند و فرانت اند وبسایتا، نه سیستم های اینترنتی، بازی و سیستم عامل و نمیدونم چی چی، واسه اونایی‌ست که میخوان بستر این کارا رو فراهم کنند.

معماری کامپیوتر:

خب قبلا راجع به مدار منطقی توضیح دادیم.
عملا میشه گف که ادامه منطقی، در معماری کامپیوتره که بهش Computer Architecture یا همون CA هم میگن.
همون طور که میدونیم انتهای درس منطقی ما یاد میگیریم که چجوری یه سخت‌افزار ساده تهیه کنیم که کارهایی که میخوایم رو انجام بده.
حالا ادامه این بحث توی معماری اینجوری پی‌گیری میشه که بعد از یه سری مقدمات راجع به سیستم اعداد باینری می‌رسیم به طراحی پروسسور ها بر پایه Mips
اصل درس پیاده سازی این سخت افزار به صورت single cyle, multicycle و pipeline هستش که به طور مفصل تو طول ترم بهش پرداخته میشه.

مدار منطقی:

اگر زیاد اهل سرک کشیدن در اجزای کامپیوتر باشید (که به احتمال خیلی زیاد هستید و اگر غیر از این بود سمت برق نمی‌رفتید) شنیده‌اید که بخش بزرگی از اجزای مختلف کامپیوتر مثل RAM یا CPU از تعداد زیادی قطعه‌ی ساده به نام «ترانزیستور» ساخته شده است. به احتمال خیلی زیاد از دوران پیش از دبیرستان به یاد دارید که ترانزیستور اصلاً قطعه‌ی پیچیده‌ای نیست. پس چطور ممکن است که اجتماع این قطعات ساده به ساخت CPU ها یا RAM های خفن و پیش‌رفته که کارهای فوق‌العاده‌ای برای ما انجام می‌دهند منجر شود؟ این دقیقاً همان مسئله‌ای است که درس مدار منطقی به حل آن می‌پردازد.
در این درس ابتدا یاد می‌گیرید با کنار هم چیدن تعداد کمی ترانزیستور قطعاتی پیشرفته‌تر بسازید که بتوانند اعمال ابتدایی مثل جمع یا ضرب را انجام بدهند. سپس یاد می‌گیرید با ترکیب این قطعات نیمه‌پیشرفته قطعات پیش‌رفته‌تر بسازید که می‌تونن عملیات پیچیده‌تری انجام بدهند و این روند تا جایی ادامه پیدا می‌کند که نهایتاً به قطعه‌ای برسید که کارهای نسبتاً بزرگ و قابل‌توجهی را انجام بدهد. همان‌کارهایی که در ابتدا سخت و مشکل به‌نظر می‌رسیدند.
بگذارید برای این که روند بالا بیشتر برایتان ملموس شود، بیشتر توضیح بدهیم:
ابتدای درس با ترانزیستور و ویژگی‌های آن آشنا می‌شوید. بعد کم‌کم یاد می‌گیرید که چگونه آن‌ها را به یک‌دیگر وصل کرده و واحدهای کوچکی به نام Gate بسازید که می‌توانند عملیات خیلی ساده‌ای روی ورودی‌هایشان انجام بدهند. با استفاده از Gate ها و روش‌های مختلفی که در درس یاد می‌گیرید به‌ترتیب به ساختارهایی می‌رسید که قدرتمند شده و می‌توانند واقعاً کارهای پیچیده‌تری انجام بدهند.
از این‌جا به بعد مسائل هر چقدر هم که پیچیده بشوند نهایتاً با ترکیب قطعات و مدارهایی که ساخت آن‌ها را یاد گرفتید، حل خواهند شد!

ساختمان‌های داده و الگوریتم در مهندسی برق:

در انگلیسی به این درس Data Structure می‌گویند و به همین اعتبار بین دانش‌جوها به DS معروف شده است! نکته‌ی جالب این‌جاست که با وجود معروف شدن به اسمDS ، این درس بیشتر حول محور الگوریتم‌ها می‌چرخد. در واقع DS در سیستم‌های دیجیتال یک پکیج فشرده از ساختمان‌داده و طراحی الگوریتم (DA) بچه‌های کامپیوتری و یک‌سری مباحث خاص رشته‌ی برق است. به‌طور کلی بخش مربوط به ساختمان داده در این درس حجم زیادی را نمی‌گیرد و بیشتر حجم درس به معرفی الگوریتم‌های مختلف از جاهای مختلف مربوط می‌شود. همین نزدیکی درس با مباحث کامپیوتری باعث می‌شود که در طول ترم تمرینات کامپیوتری هم داشته باشید که برای تحویل آن‌ها معمولاً باید از زبان‌های Python و Java استفاده کنید. اگر که از قبل این زبان‌ها را بلد باشید که عالی است اما اگر بلد نیستید هیچ جای نگرانی نیست چون توی درس مبانی کامپیوتر زبان C را یاد می‌گیرید و به همین خاطر یادگیری زبان‌های جدید برایتان اصلاً سخت نخواهد بود. البته توصیه می‌شود که برای این درس Pythonرا هم یاد بگیرید چرا که برای درس Core هم به کار خواهد آمد!

طراحی سیستم‌های نهفته مبتنی بر هسته:

این درس که معمولاً آن را Core صدا می‌زنند بیشتر به‌عنوان دنباله‌ی معنوی معماری کامپیوتر و مدار منطقی به‌حساب می‌آید. اگر بخواهیم Core را با درس‌های دیگر مقایسه کنیم بهترین تعبیر این خوهد بود: موضوعی در درس معماری کامپیوتر به اسم «پایپ‌لاین پنج مرحله» وجود دارد که یک روش معروف برای طراحی پردازشگرهاست. (اگر متوجه اسمش نشدید خیلی نگران نباشید!) این موضوع در Core تبدیل می‌شود به یک پردازنده‌ی پیچیده‌تر .

با توجه به چیزی که بالاتر گفتیم، روی هم رفته Core حالت پیش‌رفته یا حداقل نیمه پیش‌رفته‌ی معماری کامپیوتر است و از لحاظ علمی یک درس ارشد به حساب می‌آید. همین ویژگی ممکن است باعث شود Core به مذاق بعضی‌هایی که بیشتر دنبال درس‌هایی با کاربرد روزمره هستند، خوش نیاید!

اگر بخواهیم درمورد وجهه‌ی مثبت Core هم بگوییم، این درس بهشت عاشقان سخت‌افزار است. کسانی که دوست دارند بفهمند که یک پردازنده‌ی امروزی دقیقاً چطور کار می‌کند و احتمالاً اگر پای حرفشون بنشینید اسنپ‌دراگون و AMD و Intel از دهنشون نمی‌افته!

با این حال همان‌طور که قبل‌تر گفتیم ممکنه این حجم از تخصصی بودن به مذاق خیلی‌ها خوش نیاید و با این حساب Core درسی است که یا عاشقش خواهید شد یا از ابتدا منتظر رسیدن امتحانات پایان‌ترم!