

فرید فولادی-98243045

عرفان رفیعی اسکویی-98243027

1. **میکرو پراسسور ها** : تمام آن چه با سی پی یو در ارتباطه مثل دیتا پث و واحد کنترل و ALU و رجیستر و ... همه تجمیع بشن روی یک چیپ. حتی اگر حافظه ای نباشه مثال رم و رام نباشه بازم چیپ میشه ریز پردازنده.

میکرو کامپیوتر : از یک سی پی یو تشکیل شده که داده پردازش و تحلیل میکنه و معمول به صورت یک میکروپروسسور یک چپیی هست. ولی میکروپراسسور خالی کافی نیست چون باید با دنیای بیرون در تعامل باشه تا تبدیل به کامپیوتر بشه - بر < کامپیوتریکه مبت یک میکروپراسسور هست میگیم میکرو کامپیوتر. یعنی سی پی یو بهش حافظه وصل هست مثل رم یا رام که ورودی و داده را بخونه و بنویسه و یک سری پورت ورودی و خروجی داره که با دنیای خارج مثل ادوات جانبی، کاربر، سنسور ها و حسگر ها و فعالگر ها و برد و دیسک و شبکه و ... در ارتباط هست. ی ک گذرگاه هم الزم داریم به اسم bus که ای ن اجزای مختلف را به هم وصل کنه و بینشون ارتباط برقرار کنه. پس سی پی و از جنس میکرو پراسسور و حافظه و پورت های آی او که با دیوایس های خارجی آی او در ارتباط هستن و باس اجزای تشکیل دهنده آن هستن.

میکرو کامپیوتر سیستم : وقتی میگیم سیستم ینی ترکیب سخت افزار و نرم افزار پس منظور اینجا میکرو کامپیوتری است که شامل رجیستر ها و ALU و ریز پردازنده و ... است. طبق تعریف یک ریز کامپیوتر یا میکرو کامپیوتر داریم که علاوه بر سی پیو و این چیزا حافظه های رم و رام و پورت های O/I داره و بعد یک سامانه ریز کامپیوتر داریم که علاوه بر سخت افزار ریز کامپیوترمون ادوات ورودی خروجی و دیوایس های O/I مثل مانیتور وصل به کارت گرافیت و ... داره و نرم افزار های سیستمی و کاربردی هم داره. کل این می شه لپ تاپ ما + نرم افزار و همه چی که روش نصبه = سیستم ریز کامپیوتری یا همان میکرو کامپیوتر سیستم.

میکروکنترلر : به نوعی یک کامپیوتر است طوری که کل این کامپیوتر نه فقط سی پی یو اش روی یک چیپ جا شده باشه. تفاوت اصلی آن با یک میکروپراسسور این است که در میکروپراسسور فقط ALU و رجیستر ها و کنترل یونیت روی چیپ هستن ولی در می کروکنترلر علاوه بر سی پی یو که شامل این سه تا ها هست اجزای دیگه مثل رم و رام و یک سری ادوات جانبی و پورت های O/I هم روش قرار دارن.

2. دوتا فلسفه متفاوت برای طراحی پردازنده ها هست : پردازنده مبتنی بر فلسفه **سیسک** یا **معماری های پر دستور** و کامپیوتر های مبتنی بر **ریسک** یا **معماری های کم دستور**.

ز اونجایی که برنامه های سطح بالا مثل C نبود و باید اسمبلی نوشته میشدن و حافظه هم محدود بود دیدگاه این بود که پردازنده ای داشته باشیم که دستوراتی داشته باشه که یک مرتبه کار چند تا دستور را انجام دهد یا به قوی دستورات پیچیده به پردازنده اضافه کنیم که برنامه نویس به راحتی بتونه استفاده کنه. معمول به دلیل پیچیدگی، کد کردن آن ها به زبان ماشین پیچیده است و طول دستورات متغیره از یک تا n ورد و زمان اجرای دستورات هم مختلفه و دستور العمل ها چندین فرمت و قالب کد کردن در قالب زبان ماشین دارن و خیلی وقت ها فلسفه تجاریشن اینه که با نسخه قبلی خود سازگار باشند.

ریسک ها اینطورین که بهتره ما بتونیم یک هسته پردازشی سریع و چابک طراحی کنیم به شکلی که اشکال نداشته باشه یک عملیات رو با تعداد اینستراکشن بیشتری انجام بدیم ولی همون اینستراکشن هارو با سرعت بیشتری انجام بدیم. کامپایلر عملیات سطح بالا رو میشکته به دستورات بیشتر. معمول طول برنامه های ریسک بیشتره ولی در مجموع فرکانس کارش بیشتره و برنامه هارو سریع تر اجرا میکنه.

در ریسک طول دستورات ثابت و زمان برای بسیاری از اینستراکشن ها ثابت و تکنیک های پیاده سازی برای اجرا و کارایی بالا مثل پایپ لاین و ... سریع تر اجرا میشن.