

هم طراحی سخت افزار و نرم افزار

جلسه دوم: مقدمات

ارائه دهنده: آتنا عبدی

a_abdi@kntu.ac.ir



مقدمه‌ای بر هم‌طراحی سخت‌افزار نرم‌افزار

- مفهوم هم‌طراحی سخت‌افزار و نرم‌افزار در حیطه طراحی سیستم‌های دیجیتال مطرح می‌شود

- سیستم: مجموعه‌ای از اجزا که برای هدف واحدی باهم کار می‌کنند

- سیستم از دید کاربر: رعایت الزامات عملیاتی و غیرعملیاتی مشخص



- سیستم از دید طراح: اجزای سخت‌افزاری و نرم‌افزاری

- سخت‌افزار: تسریع سرعت اجرا

- نرم‌افزار: قابل اجرا روی بستر سخت‌افزاری



HW



SW



مقدمه‌ای بر هم‌طراحی سخت‌افزار نرم‌افزار

• الزامات هر سیستم

• کارکردی (Functional): خروجی تابعی از ورودی باشد

• سیستم واسط گرافیکی داشته باشد، قابلیت احراز هویت داشته باشد، فعال شدن هشدار و ...

• غیرکارکردی (Non-Functional): چگونگی انجام عملیات سیستم

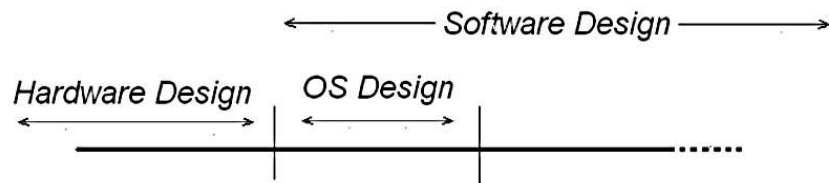
• سرعت اجرا، قیمت، سادگی استفاده، توان مصرفی و





مقدمه‌ای بر هم‌طراحی سخت‌افزار نرم‌افزار

- روال طراحی سیستم‌های دیجیتال
- بهبود تکنولوژی ساخت تجهیزات با هدف
- افزایش کارایی، کاهش هزینه‌های ساخت و تولید، سادگی برنامه‌ریزی
- تحقق این اهداف در طراحی، به هر دو نوع اجزای سخت‌افزاری و نرم‌افزاری وابسته است
- از دیدگاه کارکردی هر دو بخش به یک اندازه مهم است و تفاوت در رویکرد عملکردی‌شان می‌باشد



سیستم‌های دیجیتال



- هدف و تمرکز ما در این درس روی سیستم‌های نهفته
- تجهیزات محاسباتی تعبیه شده در سایر دستگاه‌های الکترونیکی
- برای دنبال کردن هدف مشخصی در سیستم موردنظر تعبیه می‌شود (کنترل، تعامل)
- Perform a Specific Task

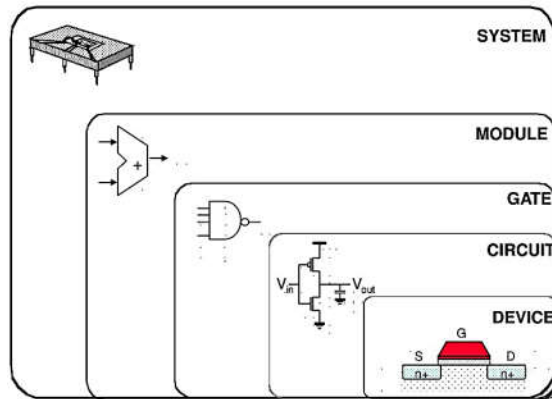
- درصد زیادی از تولیدات سیستم‌های دیجیتالی در این دسته قرار دارند



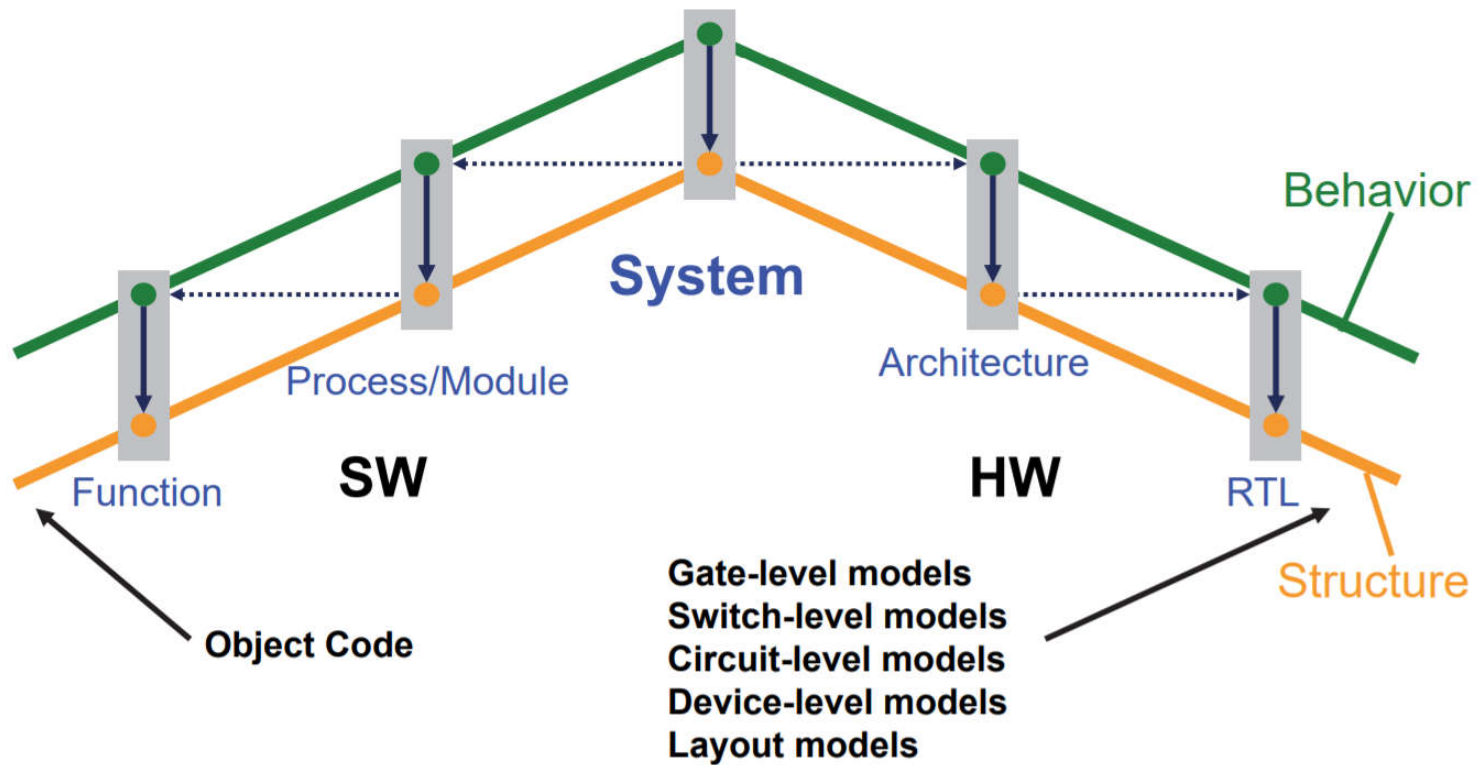


سطوح تجرید طراحی

- تمرکز ما در این درس بر طراحی سطح سیستم
- سطح بالای طراحی است، سیستم = اجزای سخت‌افزاری + ماژول‌های نرم‌افزاری
- هدف کارکرد سیستم است و جزئیات پیاده‌سازی اهمیت ندارد
- تقسیم طراحی به بخش‌های سخت‌افزاری و نرم‌افزاری با هدف رعایت الزامات غیرکارکردی



سطوح تجرید طراحی





مفهوم سخت‌افزار در فرایند هم‌طراحی

- سخت‌افزار مفهوم عام دارد و طرح‌های سخت‌افزاری

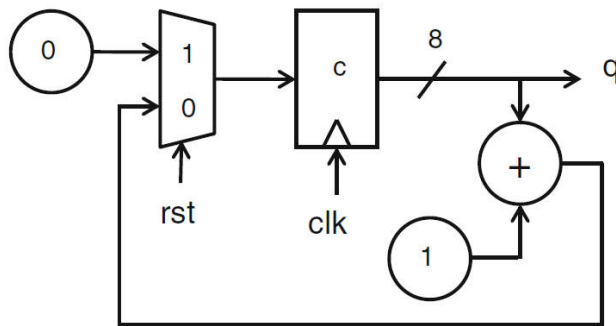
- سرعت و هزینه بالا

- هدف ما در این درس

- مدار دیجیتال سنکرون

- مدلسازی در سطح RTL

- بکارگیری به‌صورت ماژول کاربرد خاص





مفهوم نرم افزار در فرایند هم طراحی

- نرم افزار مفهوم عام دارد و طرح های نرم افزاری
- انعطاف پذیری بالا و سادگی در پیاده سازی الگوریتم با زبان سطح بالا
- هدف ما در این درس

```
#include <stdio.h>
void main()
{
    int first, second, add, subtract, multiply;

    clrscr();
    first=22;
    second=10;

    add = first + second;
    subtract = first - second;
    multiply = first * second;
```

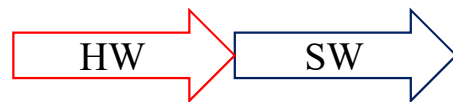
- طراحی واسطه های سخت افزار/نرم افزار
- تمرکز بر جزئیات سطح پایین نرم افزار با هدف مدیریت کارایی و هزینه
- مدلسازی نرم افزار:
- برنامه ترتیبی پیاده سازی شده در زبان سطح پایین مانند C یا اسمبلی



مقدمه‌ای بر هم‌طراحی سخت‌افزار نرم‌افزار

- رویکردهای طراحی سطح سیستم

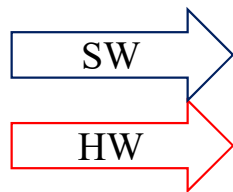
- طراحی سنتی سیستم



- سخت‌افزار و نرم‌افزار دو جزء مستقل هستند

- طراحی نرم‌افزار پس از سخت‌افزار

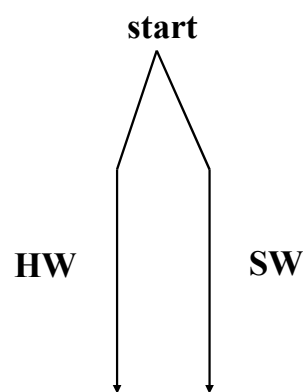
- طراحی هم‌روند سیستم



- طراحی توأم و هم‌روند سخت‌افزار و نرم‌افزار

- با هدف تحقق موثرتر اهداف عملکردی سیستم

رویکردهای طراحی سیستم



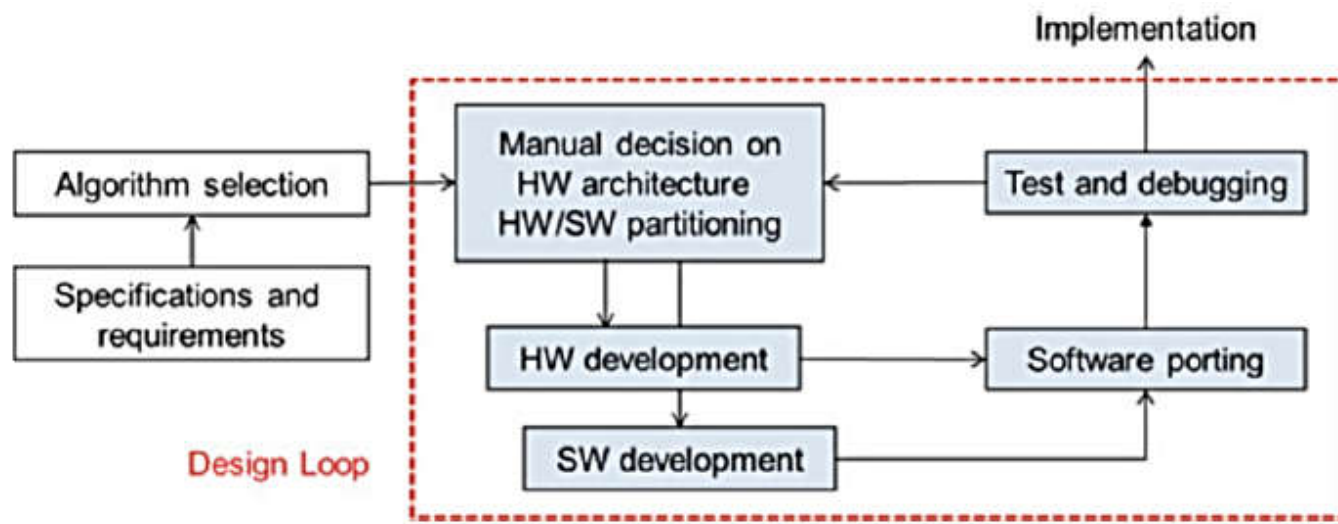
- رویکرد طراحی سنتی سیستم

- سیستم در ابتدا به دو بخش مجزای سخت‌افزار و نرم‌افزار تقسیم می‌شود
- توسعه کاملاً مجزای سخت‌افزار و نرم‌افزار
- تاثیر طراحی سخت‌افزاری بر نرم‌افزاری و برعکس دیده نمی‌شود
- تجمع دیر هنگام و محدودیت در اعمال trade-off های طراحی

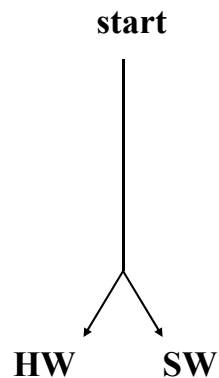
- معایب این طراحی

- هزینه تغییرات و اصلاح بسیار بالا
- کیفیت طراحی پایین (عدم تحقق برخی محدودیت‌های اولیه و کشف دیر هنگام اشکالات)
- با پیچیده شدن سیستم به شدت پیچیده می‌شود

رویکرد طراحی سنتی سیستم



رویکردهای طراحی سیستم



- طراحی هم‌روند سیستم (هم‌طراحی)

- سیستم در ابتدا به دو بخش مجزای سخت‌افزار و نرم‌افزار تقسیم نمی‌شود
- تحقق اهداف سطح سیستم و رعایت محدودیت‌ها در طراحی توأم سخت‌افزار و نرم‌افزار

- طراحی هم‌روند (concurrent)

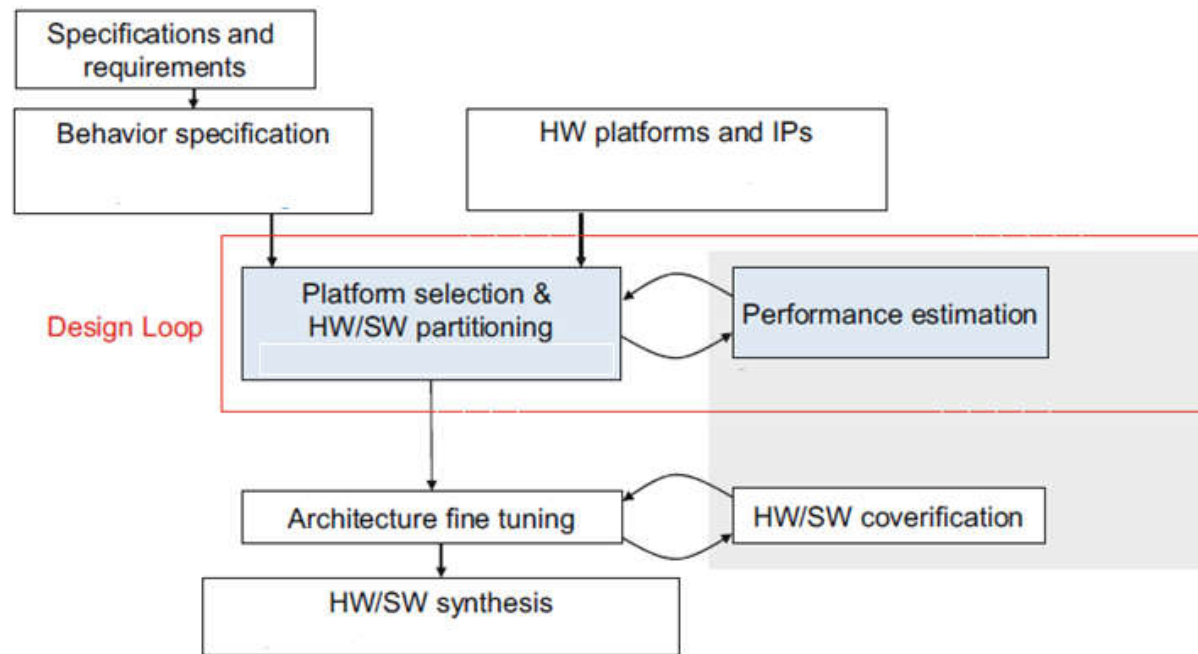
- سخت‌افزار و نرم‌افزار در یک زمان و در دو مسیر موازی توسعه داده می‌شوند

- طراحی تجمیعی (Integrated)

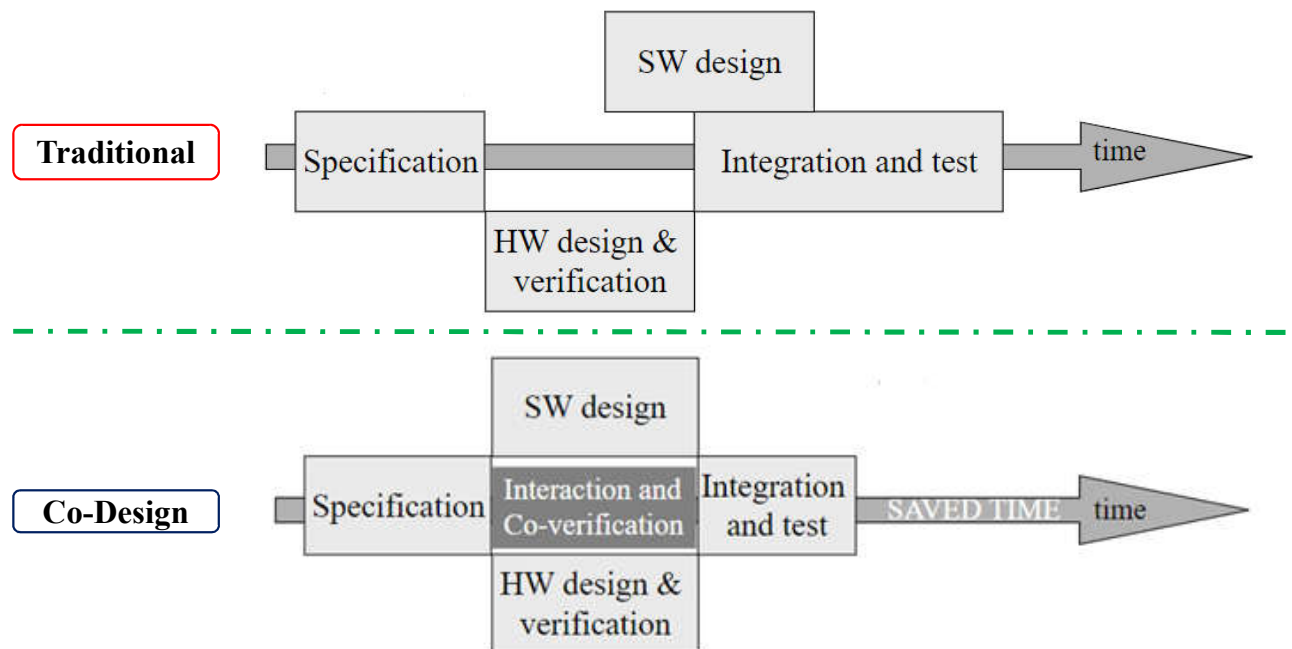
- تعامل نزدیک بین تیم توسعه‌دهنده سخت‌افزار و نرم‌افزار با هدف رعایت دقیق اهداف و محدودیت‌های طراحی



طراحی هم‌روند سیستم (هم‌طراحی)



مقایسه زمان بندی رویکردهای طراحی





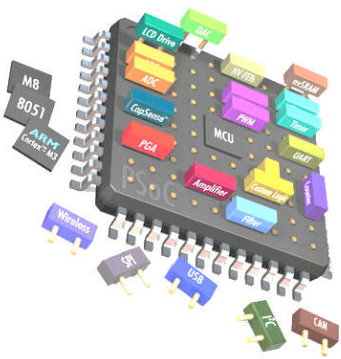
مزایای هم‌طراحی سخت‌افزار نرم‌افزار

- کاهش زمان ساخت (همکاری تیم‌های طراحی)
- امکان ارزیابی سریع‌تر نرم‌افزار توسعه داده شده
- دسترسی به طراحی موثرتر در زمان و با هزینه کمتر
- جستجوی موثرتر فضای طراحی و ایجاد تعادل بین سخت‌افزار و نرم‌افزار
- رفع سریع اشکالات حین طراحی قبل از فرایند جمع و ساخت
- تحقق ساده‌تر محدودیت‌ها و اهداف طراحی
- توان مصرفی، اندازه، زمان‌بندی، قابلیت اطمینان، ایمنی و

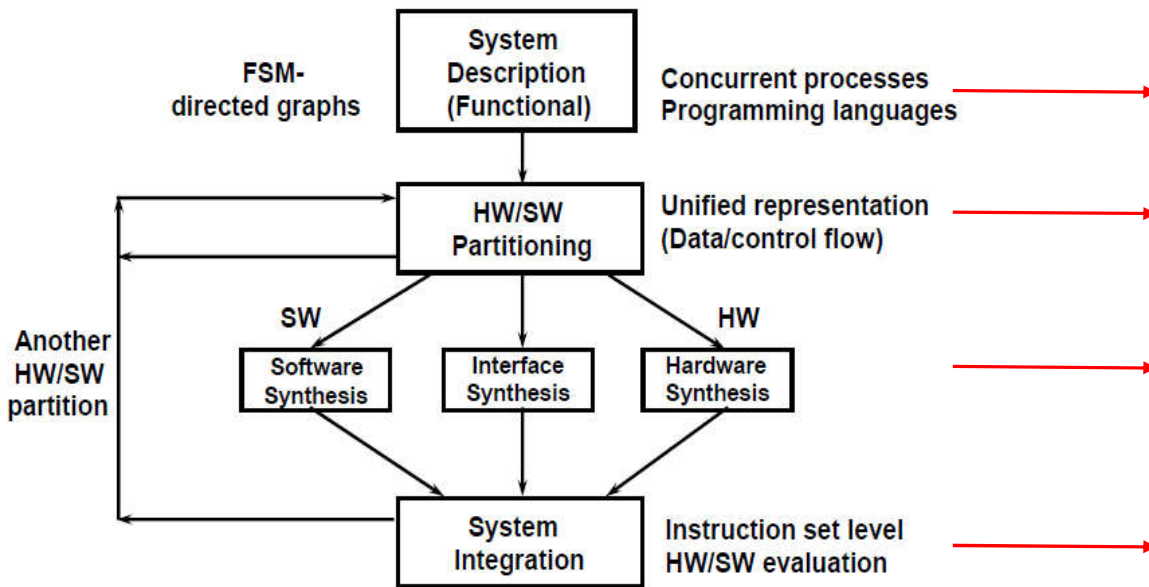


هم طراحی سخت افزار و نرم افزار

- مهم ترین کاربرد شیوه های هم طراحی سخت افزار و نرم افزار
- طراحی سیستم های نهفته (embedded systems) و سیستم های روی تراشه (SoC)
- این سیستم ها برحسب کاربرد، نیاز به طراحی دقیق دو بخش سخت افزار و نرم افزار دارند
- طراحی این سیستم ها مستلزم رعایت محدودیت هایی در کارایی، هزینه و دیگر پارامترهاست
- رعایت این محدودیت ها در شیوه طراحی توأم در سخت افزار و نرم افزار ساده تر است



روال معمول طراحی توأم



توصیف در قالب زبان‌های قابل اجرا مانند System C یا ترکیبی از HDL و C++/Java

افزار توصیف عملیاتی توسط ابزارها (profiling tools) یا روش‌های تخمینی و مشخص شدن زمان اجرای هر بخش و گلوگاه‌ها و نقاط بحرانی طراحی (مسئله NP-hard)

طراحی موازی سخت‌افزار، نرم‌افزار و واسط توسط تیم‌های مجزا و اعتبارسنجی بر بستر مشترک (cosimulation)

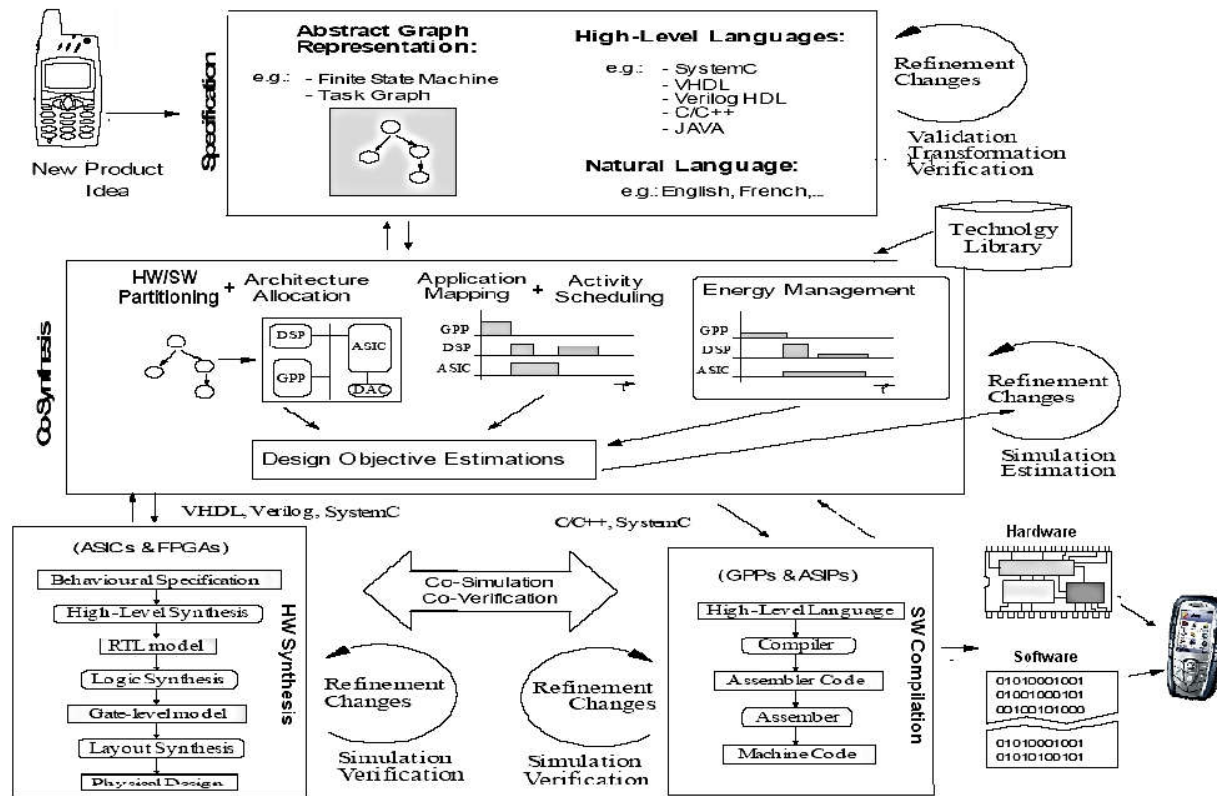
تجمع واحدهای طراحی شده و بررسی تحقق محدودیت‌ها در صورت برآورده نشدن محدودیت‌های سیستم، افزار دیگر بررسی شده و بدین ترتیب فضای طراحی جستجو می‌شود

روال معمول طراحی توأم

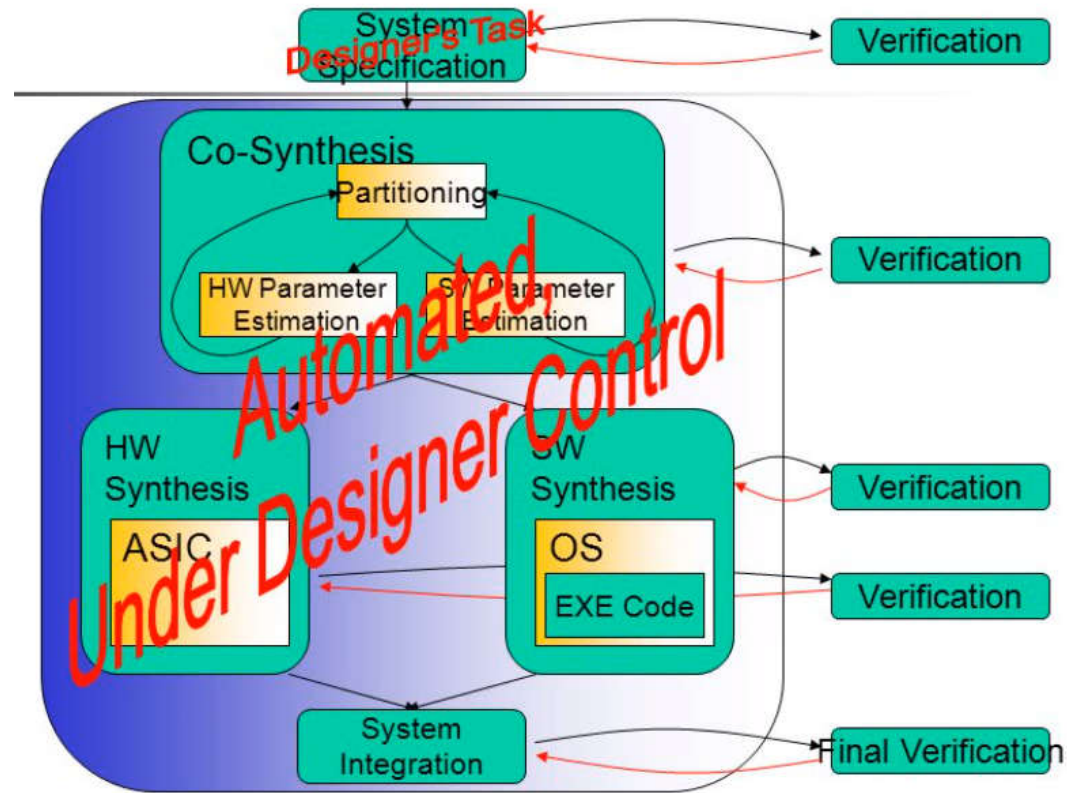


- توصیف سیستم براساس کارکرد آن، گام اولیه طراحی است
- این توصیف براساس زبان‌های مناسب یا ساخت مدل انتزاعی انجام می‌گیرد.
- نگاشت مدل انتزاعی به معماری بستر پیاده‌سازی در مرحله افزار انجام می‌گیرد
- در این گام روش‌های تخمینی به‌منظور ارزیابی افراز انجام شده و مناسب بودن آن بکارگرفته می‌شود.
- پس از افراز فرایند سنتز و ساخت نرم‌افزار و سخت‌افزار به‌صورت موازی دنبال می‌شود
- در این مرحله زمان‌بندی و شیوه ارتباط این دو بخش معین می‌شود
- در صورت نامناسب بودن خروجی این فاز، افراز دیگری به سیستم اعمال می‌شود.

روال معمول طراحی توأم



روال معمول طراحی توأم





پیش‌نیازهای محیطی برای هم‌طراحی ایده‌ال

- انتخاب شیوه نمایش و مدلسازی یکپارچه و بی‌طرف
- زبان توصیف یا مدل‌هایی که به سخت‌افزار یا نرم‌افزار متمایل نباشند
- مانند: UML، System C، مدل‌های Petri Net و Dataflow Diagrams
- پشتیبانی از روش‌های افراز بازگشتی
- تغییر و پویش افرازهای ممکن بین سخت‌افزار و نرم‌افزار با هدف جستجوی فضای طراحی
- لایه شبیه‌سازی و ارزیابی تجمیعی
- شبیه‌سازی و ارزیابی هم‌زمان سخت‌افزار و نرم‌افزار در محیط یکپارچه

مباحثی که این جلسه آموختیم



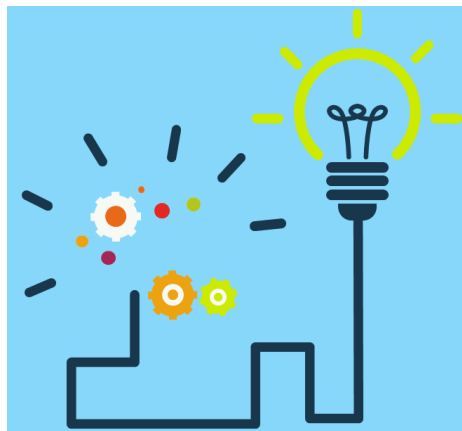
- مقدماتی بر مفهوم هم‌طراحی سخت‌افزار و نرم‌افزار

- طراحی سطح سیستم سخت‌افزار و نرم‌افزار

- رویکرد طراحی سنتی در مقابل هم‌طراحی

- کاربردهای هم‌طراحی

- روال هم‌طراحی و مراحل مهم در فرایند



مباحث جلسه آینده



- آشنایی با سیستم‌های نهفته
- مهم‌ترین کاربرد در فرایند هم‌طراحی سخت‌افزار و نرم‌افزار

