

# هم طراحی سخت افزار نرم افزار

جلسه نوزدهم: سنتز توأم-زمان بندی

ارائه دهنده: آتنا عبدی

[a\\_abdi@kntu.ac.ir](mailto:a_abdi@kntu.ac.ir)

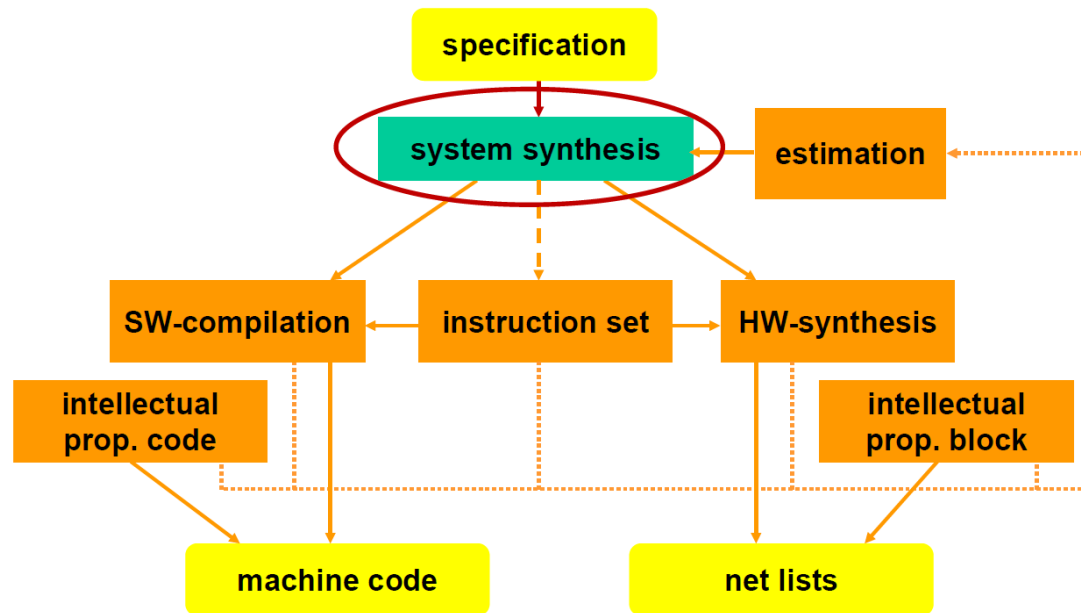
# مباحث این جلسه



- سنتز توام در روال هم طراحی سخت افزار و نرم افزار

- زمان بندی (Scheduling)

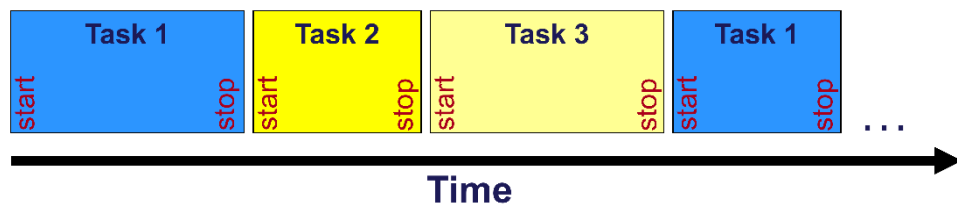
- رویکردهای زمان بندی



# فرایند زمان‌بندی



- یکی از فرایندهایی که در حین سنتز توأم انجام می‌گیرد
- مشخص کردن ترتیب زمانی اجرای عملیات بر اجزای پردازشی و منابع سیستم
- در مرحله نگاشت و پس از افراز انجام می‌گیرد
- رعایت محدودیت‌های زمان اجرا و عملکردی سیستم در صورت وجود
- مدیریت دسترسی به منابع مشترک



# فرایند زمان‌بندی (ادامه)



- در حین زمان‌بندی لازم است اطلاعات منابع موجود و پردازش‌ها را بدانیم
- تاخیر اجرای هر پروسه بر منابع پردازشی
- تخصیص پروسه‌ها و وظایف به منابع پردازشی مشخص است
- مشخص کردن ترتیب اجرا با هدف رعایت محدودیت زمان اجرا
- زمان‌بندی با دو رویکرد ایستا و پویا
- زمان‌بندی بر بستر سیستم‌های تک‌پردازنده و چندپردازنده‌ای

# فرایند زمان‌بندی (ادامه)



- ورودی‌های مرحله زمان‌بندی:
  - برنامه کاربردی در قالب گراف وظایف
  - معماری سیستم در قالب المان‌های پردازشی و اتصالات
  - خروجی پارتیشن‌بندی وظایف بر المان‌های پردازشی سیستم (تصمیم‌گیری با زمان‌بندی ابتدایی)
- خروجی‌های مرحله زمان‌بندی:
  - ترتیب اجرای وظایف بر المان‌های پردازشی
  - زمان شروع و پایان هر وظیفه

# فرایند زمان‌بندی (ادامه)



- در حین زمان‌بندی هر وظیفه در سه وضعیت می‌تواند قرار داشته باشد:

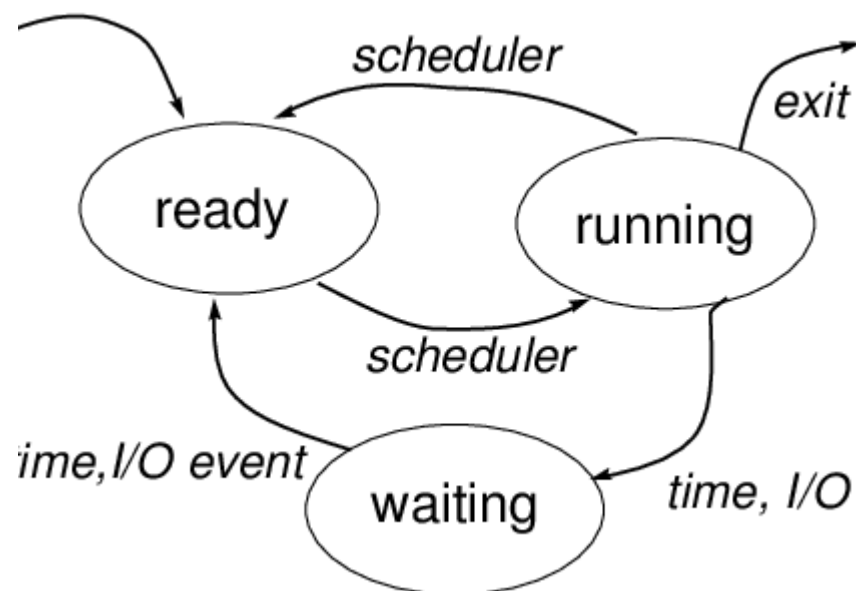
- آماده: تحقق تمامی پیش‌نیازهای اجرایی و داده‌ای

- زمان ورود وظیفه به این حالت: Release Time

- انتظار: تاخیر در اجرا یا انتظار برای یک رخداد

- در حال اجرا

- زمان ورود وظیفه به این حالت: Start Time



# الگوریتم‌های زمان‌بندی



- با فرض مشخص بودن نگاشت و تخصیص منابع پردازشی
- عدم وجود محدودیت در منابع پردازشی
- زمان‌بندی با هدف کمینه کردن زمان اجرا تا حد ممکن
  - الگوریتم زمان‌بندی ASAP
  - الگوریتم زمان‌بندی ALAP

# الگوریتم‌های زمان‌بندی ASAP



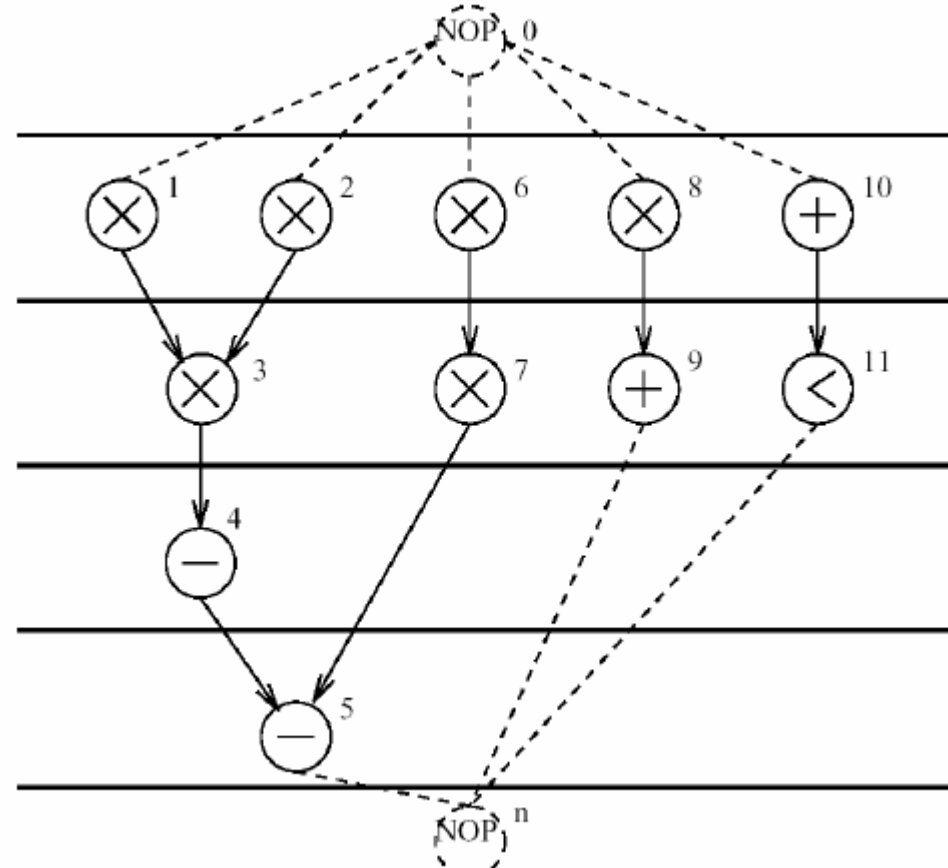
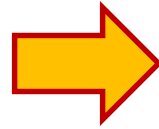
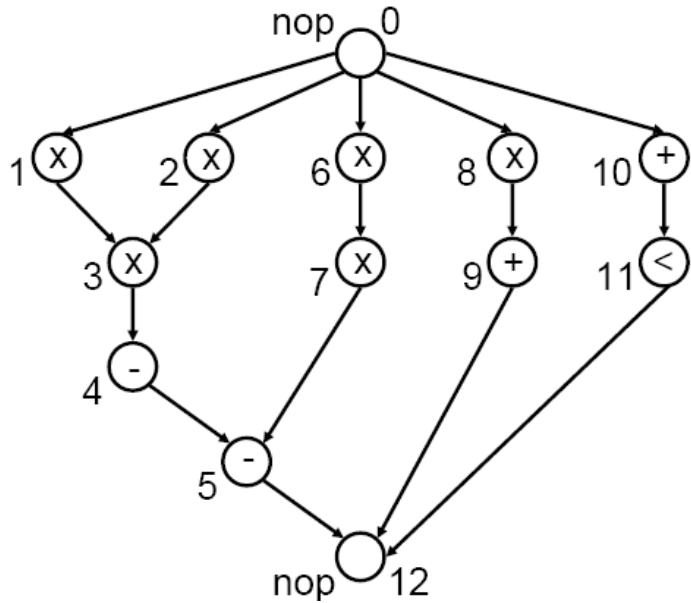
- ساده‌ترین الگوریتم زمان‌بندی با هدف کمینه کردن تاخیر اجرا (As Soon As Possible)
- تنظیم زمان شروع هر وظیفه در زودترین زمان ممکن براساس وابستگی داده‌ای آن
- اهمیت دادن به وابستگی‌های داده‌ای
- متصل کردن تمامی گره‌های بدون وابستگی به یک گره مجازی Source
- متصل کردن تمامی گره‌های انتهایی برگ به یک گره مجازی Sink
- شروع با زمان‌بندی گره‌های متصل به source در مرحله اول
- زمان‌بندی گره‌های متصل به موارد زمان‌بندی شده مرحله اول و ....





# الگوریتم‌های زمان‌بندی ASAP (مثال)

- پیاده‌سازی بر گراف نمونه



# الگوریتم‌های زمان‌بندی ALAP

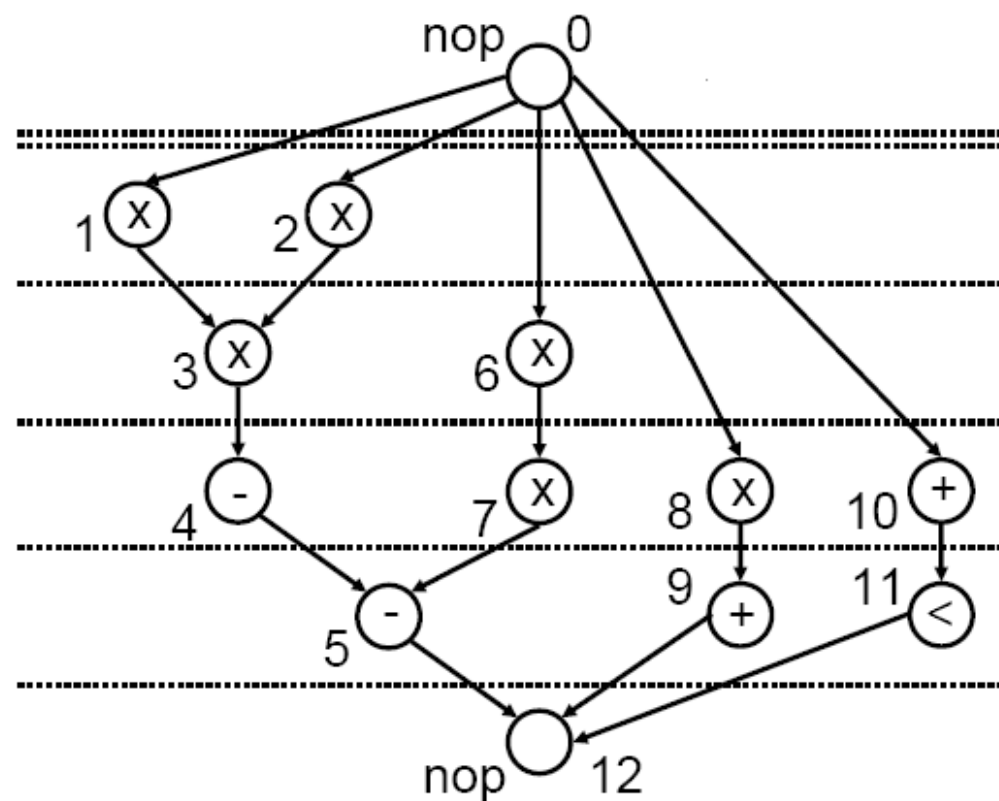
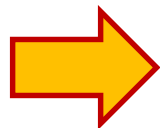
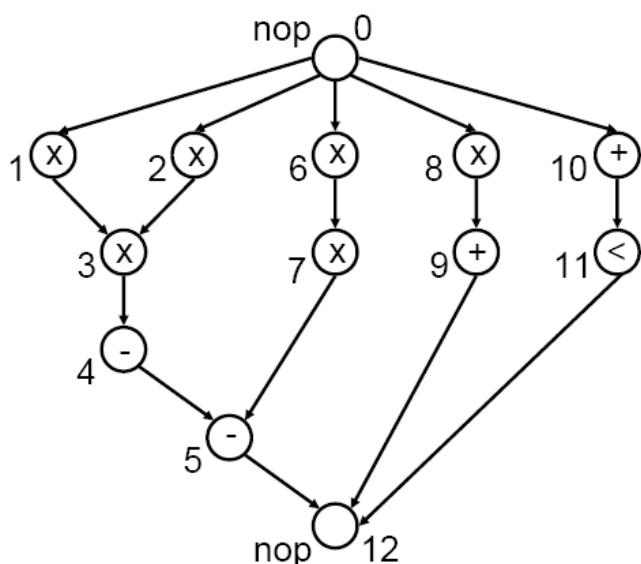


- بر خلاف ASAP تعریف می‌شود (As Late As Possible)
- با داشتن حد زمانی اجرا، وظایف را در دیرترین زمان ممکن اجرا کنیم
- مشابه ASAP ولی در اینجا از انتها به ابتدا مرتب می‌کنیم
- در اینجا هم گره Sink و Source داریم
- با زمان‌بندی هر گره در مرحله بعد، predecessor های آن زمان‌بندی می‌شوند
- شروع از گره‌هایی که مستقیم به Sink وصل هستند

# الگوریتم‌های زمان‌بندی ALAP (مثال)



• پیاده‌سازی بر گراف نمونه

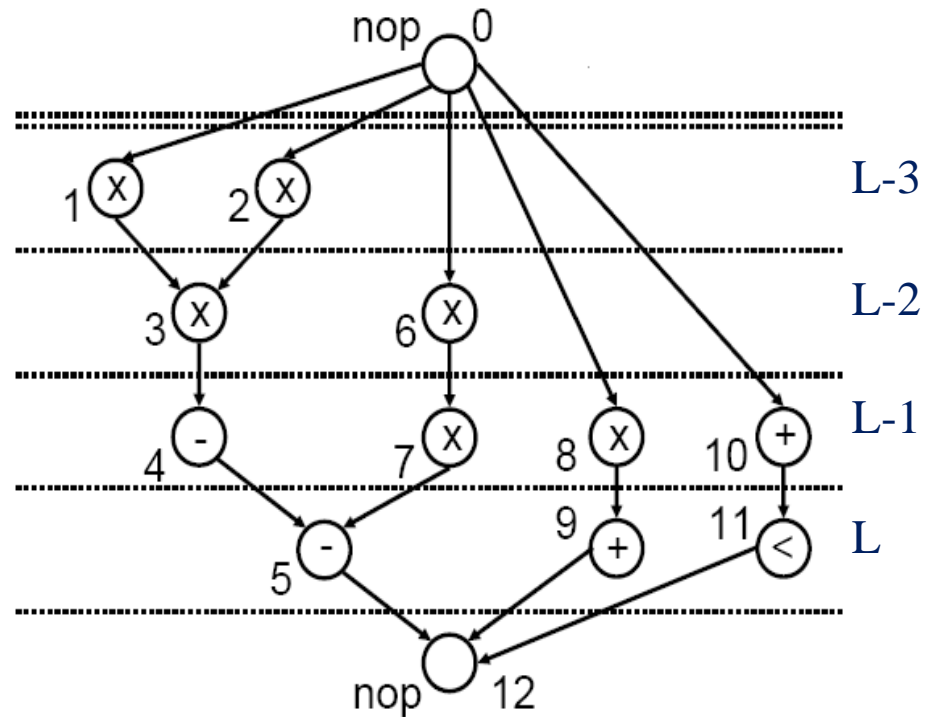
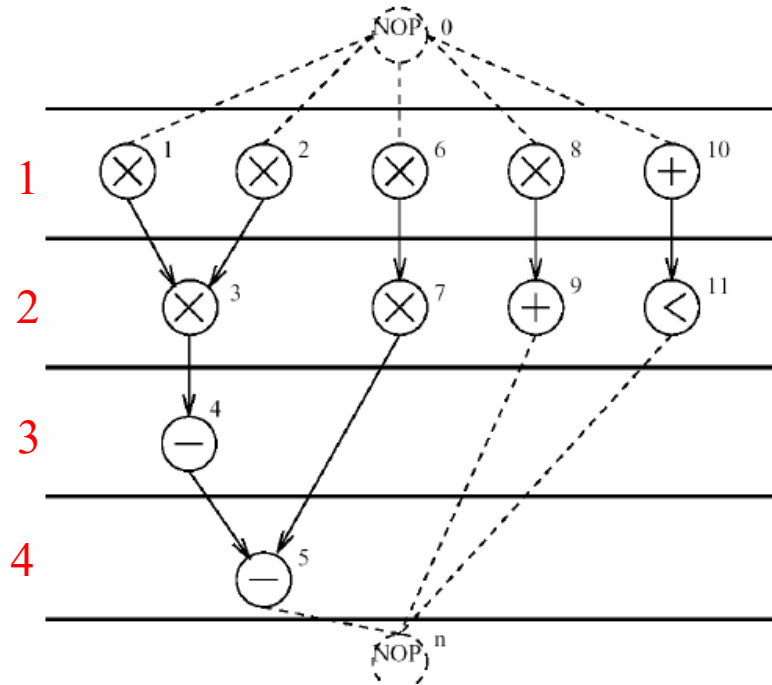


# اهمیت زمان‌بندی با روش‌های ASAP و ALAP



- تفاوت جواب‌های بدست آمده برای زمان اجرا با این دو الگوریتم مهم است: Mobility
- این پارامتر نشان می‌دهد تا چه حد در زمان‌بندی قابلیت انعطاف وجود دارد
- گره‌هایی که این پارامتر برایشان صفر است مسیر بحرانی را تشکیل می‌دهند
- استفاده از این معیار با هدف تغییر تصمیم زمان‌بندی و برقرار کردن سایر محدودیت‌ها و اهداف مسئله
- تمرکز زمان‌بندی در روش‌های اتوماتیک، بر مسیر بحرانی است

# پارامتر Mobility



# زمان‌بندی با در نظر گرفتن محدودیت منابع



- محدود بودن تعداد منابع و تاثیر بر فضا و هزینه
- تعداد منابع پردازشی سیستم محدود است و سیستم بدون محدودیت واقعی نیست
- نوبت‌دهی به اجرای وظایف مبتنی بر محدودیت منابع پردازشی براساس
  - وابستگی‌های داده‌ای
  - اولویت وظایف آماده



# زمان‌بندی با در نظر گرفتن محدودیت منابع (ادامه)

- شکل‌گیری مصالحه بین تاخیر و تعداد منابع در سیستم
- روش‌های قطعی: ILP
- تعریف مسئله در قالب معادله و نامعادله و حل آن
- قابل حل برای حالت‌های محدود و مناسب برای مشخص کردن حدود مسئله
- روش‌های مکاشفه‌ای: مانند زمان‌بندی لیستی (List Scheduling)
- مبتنی بر اولویت‌دهی به گره‌های پردازشی
- در نظر گرفتن وابستگی‌های داده‌های و محدودیت‌های منابع سیستم