



طراحی توأم نرم افزار و سخت افزار، ۸۱۰۱۶۱۲

Course Name, 8101...										نام انگلیسی درس
واحد:	مهندسی کامپیوتر					مهندسی برق				نوع درس
	فناوری اطلاعات	سخت افزار	نرم افزار	دیجیتال	کنترل	پزشکی	قدرت	الکترونیک	مخابرات	
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
										اجباری
										<input type="checkbox"/>
										اختیاری
										کارشناسی <input checked="" type="checkbox"/> تحصیلات تکمیلی <input type="checkbox"/>
										مقطع
										ندارد.
										همیناها
معماری کامپیوتر (۸۱۰۱۴۲۳) - برنامه نویسی پیشرفته (۸۱۰۱۱۱۹) - مدارهای منطقی (۸۱۰۱۳۶۷)										پیش نیازها
طراحی دیجیتال سخت افزار، معماری های سیستم های کامپیوتری، زبان برنامه نویسی C، زبان توصیفی سخت افزار										مطالب پیش نیاز
[1] Patrick R. Schaumont, <i>A Practical Introduction to Hardware/Software Codesign</i> , Springer, 2010. [2] Daniel D. Gajski, S. Abdi, A. Gerstlauer, and G. Schirner, <i>Embedded System Design: Modeling, Synthesis and Verification</i> , Springer, 2010.										کتاب های مرجع
این درس مقدمه ای بر طراحی سیستم های نهفته ی الکترونیکی که از سخت افزار علاوه بر نرم افزار استفاده میکنند، همراه با تأکیدی بر متدولوژی های طراحی مدرن و ناحیه ی انتقال بین سخت افزار و نرم افزار را فراهم میکند. این شامل مطالعه ای از روش های مدل سازی اجزای سخت افزار و نرم افزار در سطح های مختلف از مجرد سازی و مطالعه ای بر روش های رابط بین اجزای سخت افزار و نرم افزار است.										اهداف درس
دانشجویانی که این درس را با موفقیت پشت سر بگذارند قادر خواهند بود										نتایج درس
۱- به آنالیز و شرح روند کنترل و روند داده های یک برنامه ی نرم افزاری و یک توضیحات سخت افزاری مبتنی بر چرخه. ۲- به تبدیل برنامه های نرم افزاری ساده به توضیحات سخت افزاری مبتنی بر چرخه با رفتار یکسان و برعکس. ۳- بخش بندی برنامه های نرم افزاری ساده به اجزای سخت افزاری و نرم افزاری و ساخت ساخت رابط های مناسب سخت افزار-نرم افزار برای انعکاس این بخش بندی. ۴- شناسایی گلوگاه های کارایی در یک معماری سخت افزار-نرم افزار و بهینه کردن آن ها با دگرگونی های روی اجزای سخت افزاری و نرم افزاری. ۵- استفاده از نرم افزار شبیه سازی برای شبیه سازی توأم برنامه های نرم افزاری به همراه توضیحات سخت افزاری مبتنی بر چرخه.										
۱. مقدمه و طبیعت سخت افزار و نرم افزار (خصوصیات اساس سخت افزار و نرم افزار و انگیزه های طراحی توأم سخت افزار/نرم افزار) ۲. چالش های طراحی سطح سیستم و متدولوژی ها (بررسی کردن مفاهیم ابتدایی و اصول تکنیک های طراحی سیستم و متدولوژی ها، شامل نرم افزار و سخت افزار که صریحا در سیستم های نهفته ی امروزه که نرم افزار و سخت افزار تبادل پذیر است، مورد نیاز است).										فهرست مباحث



<p>۳. مدل سازی روند داده و روند کنترل (معرفی روند داده و روند کنترل به عنوان خصوصیات اساسی مشترک از سخت افزار و نرم افزار)</p> <p>۴. برنامه نویسی میکرو (معرفی معماری های micro-programmed که هنوز بسیار مشابه ماشین های RTL هستند، ولی دارای کنترلر قابل انعطاف هستند که به آنها اجازه میدهد تا با نرم افزار دوباره برنامه ریزی شوند)</p> <p>۵. هسته های نهفته ی همه-منظوره (مرور هسته های RISC نهفته ی همه-منظوره که قلب سیستم های سخت افزار/نرم افزار معمول امروزه هستند)</p> <p>۶. سنتز سیستم (مقابله با مسائل و راه حل های احتمالی در سنتز و درستی یابی نرم افزار و جز سخت افزار مورد نیاز در یک پلتفرم سیستم نهفته)</p> <p>۷. سیستم روی تراشه (مرتبط کردن هسته ی نهفته ی همه منظوره به FSMD در زمینه ی معماری System-on-chip)</p> <p>۸. باس های on-chip (مطرح کردن یک ساختار معمولی باس روی تراشه و توضیح چگونگی انتقال اطلاعات به صورت بهینه بین سخت افزار و نرم افزار)</p> <p>۹. رابط های سخت افزار/نرم افزار (۳ جا در معماری SoC را نمایش میدهد که یک طراح بتواند سخت افزار سفارشی خود را متصل کند. این شامل memory-mapped interface و، رابط coprocessor و مسیر داده ی پردازشگر سفارشی میشود)</p> <p>۱۰. ترکیب با استفاده از پلتفرم سخت افزار قابل تنظیم (معرفی یک فریمورک برای ترکیب سخت افزار-نرم افزار در سطح بالا و تولید برنامه)</p> <p>۱۱. ابزارهای تجزیه و تحلیل نرم افزار برای طراحی ASIP (نمایه سازی نرم افزار و ابزارها و روش های شخصی سازی مجموعه دستورالعمل ها)</p> <p>۱۲. طراحی پوسته ی کنترل coprocessor (نشان میدهد که چگونه یک طراح میتواند یک ماژول دلخواه سخت افزاری را به یکی از ۳ رابط سخت افزار/نرم افزار متصل کند)</p>	
<p>زبان برنامه نویسی C و زبان توصیفی سخت افزار</p>	<p>نرم افزارها و ابزارهای مورد نیاز</p>
<p>۱۰ تکلیف</p>	<p>تکالیف پیشنهادی</p>
<p>۴ تمرین کامپیوتری</p>	<p>پروژه های پیشنهادی</p>
<p>تکالیف ۱۰٪ کوئیز و تمرین های کامپیوتری ۲۰٪ امتحان میان ترم ۳۰٪ امتحان پایان ترم ۴۰٪</p>	<p>نمره دهی پیشنهادی</p>
<p>[1] Jingzhao ou and Viktor k. PraSanna, <i>Energy Efficient Hardware-Software Co-Synthesis Using Reconfigurable Hardware</i>, CRC Press, 2010. [2] K. Karuri and R. Leupers, <i>Application Analysis Tools for ASIP Design</i>, Springer 2011.</p>	<p>سایر مراجع</p>



<p>[3] Ivan Radojevic and Z. Salcic, <i>Embedded Systems Design Based on Formal Models of Computation</i>, Springer 2011.</p> <p>[4] Giovanni De Micheli, R. Ernst, and W. H. Wolf, <i>Readings in hardware/software co-design</i>, Morgan Kaufmann, 2002.</p> <p>[5] Wayne Hendrix Wolf, <i>Computers as components: principles of embedded computing system design</i>, 2nd Edition, Morgan Kaufmann, 2008.</p>	
	نام استاد تنظیم کننده شرح درس
	تاریخ تنظیم شرح درس