



به نام خدا
تمرین سری اول
تئوری اعداد - معماری کامپیوتر
استاد : دکتر محمد رضا پورفرد
تدریس یار : جواد سبحانی
دانشگاه صنعتی امیرکبیر، دانشکده مهندسی برق
پاییز ۱۴۰۴



- فایل حل خود را به همراه سایر فایل های مربوطه به صورت فایل فشرده (*Zip or RAR*) آپلود کنید.
- فزمت فایل های فشرده باید به صورت زیر باشد :

HW1_StudentNumber

- تمرین های اضافی برنامه نویسی باید در زبان *VHDL (Verilog)* انجام شود.
- گزارشی از عملکرد کد های نوشته شده تهیه کرده و به همراه فایل های کد آپلود کنید.
- در صورت بروز اشکالی در آپلود تمرین در *Courses* از طریق ایمیل *j.sobhani@aut.ac.ir* فایل ها را ارسال کنید و یا در تلگرام اعلام کنید.

۱ نمایش اعداد

۱-۱ تبدیل بیس

۱. اعداد زیر را به بیس های خواسته شده تبدیل کنید.

$$(00110011)_2 = (?)_{16} \quad ۱.$$

$$(2445)_8 = (?)_{10} \quad ۲.$$

$$(-35.625)_{10} = (?)_2 \quad ۳.$$

۲. *Bonus*: تحقیق کنید چرا عدد 0.1 را نمی توان به صورت دقیق در سیستم *Floating point*

نشان داد

۲-۱ نمایش اعداد

۲. عدد دوازده بیتی 110011001101 را در نظر بگیرید و موارد زیر را محاسبه کنید.

۱. *Unsigned Integer*

۲. *Sign Magnitude*

۳. *1's Complement*

۴. *2's Complement*

۲. *Bonus*: کد *VHDL(Verilog)* بنویسید که یک عدد *integer* دریافت کرده و *2's Complement* و یا *2's Complement* را محاسبه کند. انتخاب بین آن دو با یک بیت *Select* انجام می شود.

۳-۱ *Sign Extension*

۱. اعداد علامت دار زیر در سیستم (*2's Complement*) را *Sign Extend* کنید.

۱. $(1001010110)_2 = (?)_2$

۲. $(1A3)_{16} = (?)_2$

۳. $(01100101)_2 = (?)_2$

۴. $(245)_8 = (?)_2$

۲. *Bonus*: کد *VHDL(Verilog)* بنویسید که با دریافت یک عدد و تعداد بیت های آن، بتواند آن را به ۳۲ بیت *Sign Extend* کند. همچنین با یک بیت بین *1's Complement* و *2's Complement* انتخاب انجام شود.

۲ *Number Manipulation*

۱-۲ *Bit – Level Operation*

۱. عملیات خواسته شده را روی اعداد علامت دار زیر اجرا کنید. (اعداد *2's Complement* هستند)

۱. $(0xFA2) \wedge (0x221) = (?)_2$

۲. $(10011101) \vee (101101) = (?)_2$

۳. با استفاده از عملیات سطح بیت، با ارزش ترین بیت عدد 1001110 را بیابید.

۴. با استفاده از عملیات سطح بیت ۴ رقم کم ارزش عدد 0x2AD را یک کنید.

۲-۲ Logic Operation

۱. عملیات خواسته شده را روی اعداد علامت دار زیر اجرا کنید. (اعداد 2's Complement هستند)

۱. $(10011101) \parallel (101101) = (?)_2$

۲. $(0xF7) \&\& (000101101) = (?)_{16}$

۳. $!(110010) = (?)_2$

۴. $!(0xFA2) \parallel !! (0x221) = (?)_2$

۳-۲ Shift Operation

۱. عملیات خواسته شده را روی اعداد علامت دار زیر اجرا کنید. (اعداد 2's Complement هستند)

۱. $(10011101) \ll 4 = (?)_2$

۲. $(110010) \gg 3 = (?)_2$

۳. با استفاده از شیفت، بیت با ارزش عدد 10010100101 را استخراج کنید.

۴. با استفاده از عملیات سطح بیت و شیفت بیت ۱۲ هم از عدد 0x21AF را یک کنید.

۲. در مورد شیفت سخت افزاری تحقیق کرده و چگونگی انجام آن در یک سیکل کلاک را مورد بررسی قرار دهید.

۴-۲ Addition

۱. جمع های خواسته شده را یکبار با فرض Unsigned و بار دیگر با فرض 2's Complement بودن محاسبه کنید (به جز دو مورد آخر).

۱. $(101101)_2 + (0x20)_{16} = (?)_2$

۲. $(0xF7)_{16} + (1535)_{10} = (?)_{16}$

$$3.14 + 1e20) - 1e - 20[\text{single precision}] \quad ۳$$

$$3.14 + (1e20 - 1e - 20)[\text{single precision}] \quad ۴$$

۲. در مورد دلیل تفاوت جواب حاصل شده در دو مورد انتهایی از سوال پیش تحقیق کنید.

Multiplication ۵-۲

۱. حاصل ضرب های زیر را محاسبه کنید.

$$(10010)_2 \times (10101)_2 = (?)_2 \quad ۱$$

$$(1010)_2 \times (1100)_2 = (?)_2 \quad ۲$$

$$13 \times (-11) \text{ using Radix} - 4 \text{ Booth encoding.} \quad ۳$$

$$(10.1)(0.1953125) \text{ in IEEE 754 single - precision.} \quad ۴$$

۲. با استفاده از الگوریتم *Booth*، حاصل $6 * 5 - 6$ را محاسبه کنید. همچنین بررسی کنید در صورت استفاده از روش معمول شیفت - جمع چند سیکل کلاک افزایش خواهیم دید.

۳. توضیح دهید چرا در معماری های جدید از *Radix - 4 SRT division* استفاده می شود.

۴. حاصل $3.625 * 1.75$ را با استفاده از متد *Q4.4* محاسبه کنید.

Division ۳

۱. حاصل $29 \div 6$ را با استفاده از *6 - bit restoring division* محاسبه کنید.

۲. حاصل $29 \div 6$ را با استفاده از *6 - bit non - restoring division* محاسبه کنید.

۳. در باره سخت افزار های مورد استفاده برای تقسیم در معماری های جدید تحقیق کنید.

Bonus Project ۴

سیستمی طراحی کنید که با گرفتن *OPCODE* و ورودی های مربوط یکی از سه عملیات جمع، تفریق یا ضرب را انجام دهد.

دیتای ورودی به صورت ۳۲ عدد ۳۲ بیتی وارد سیستم میشود. پس از اتمام بافتن پروسه محاسبه، یک *flag* به نام *OutputRdy* یک می شود.

به عنوان مثال دستور جمع ارسال میشود و دو دسته عدد *A* و *B* وارد سیستم شده و نتیجه های خروجی در *C* قرار داده خواهد شد.

لیست *OPCODE* ها:

ADD : 0110011000000000 •

SUB : 0110011001000000 •

MULT : 0110011000000001 •

تاریخ تحویل : ۱۶ آبان