



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

معماری کامپیوتر
نیمسال دوم ۱۳۹۶
تمرین چهارم



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تحويل در روز سه شنبه مورخ ۱۳۹۷/۲/۴

تمرینات خود را با فرمت studentID_studentName_HW4 در مودل آپلود کنید.

لطفا پاسخ ها خوانا و تمیز نوشته شوند.

سوالات و مشکلات خود را به ایمیل computerarchitecture2018@gmail.com بفرستید.

سوال اول

عبارت زیر را با فرض اینکه محدود به ۸ بیت هستیم و اعداد این عبارت، مکمل ۲ هستند محاسبه کنید، سپس جواب خود را به مبنای ۱۰ بازگردانید. آیا جواب شما درست است؟ چرا؟

$$5F_{16} + C2_{16}$$

ابتدا نمایش اعداد را به صورت باینری می نویسیم و جمع می کنیم. (البته می توان مستقیماً از مبنای ۱۶ تبدیل کرد)

$$5F_{16} = 0101\ 1111_2$$

$$C2_{16} = 1100\ 0010_2$$

$$0101\ 1111_2 + 1100\ 0010_2 = 0010\ 0001_2$$

توجه کنید که بیت نقلی را در نظر نگرفتیم.

جواب عددی مثبت است چون $MSB = 0$ پس می توان مستقیماً به عدد ده دهی تبدیل کرد.

$$0010\ 0001_2 = 33_{10}$$

آیا جواب درست است؟ اعداد داده شده را به دهدهی تبدیل می کنیم:

$$5F_{16} = 95_{10}$$

$C2_{16}$ عدد منفی است چون $MSB = 1$ ، پس تبدل ده دهی آن به صورت زیر است:

$$C2_{16} = -62_{10}$$

تحويل در روز سه شنبه مورخ ۱۳۹۷/۲/۴

تمرینات خود را با فرمت studentID_studentName_HW4 در مدل آپلود کنید.

لطفا پاسخ ها خوانا و تمیز نوشته شوند.

سوالات و مشکلات خود را به ایمیل computerarchitecture2018@gmail.com بفرستید.

$$95_{10} + -62_{10} = 33_{10}$$

پس جواب بالای ما درست بوده است.

جواب برای این درست است چون در بازه ی ۸ بیتی مکمل ۲ قرار دارد. ۸ بیت مکمل ۲ در بازه ی - ۱۲۸ تا ۱۲۷ قرار می گیرد و وقتی یک عدد مثبت رو با یک عدد منفی جمع می کنیم حتما در این بازه قرار خواهد داشت.
دقت کنید اگر دو عدد منفی یا هر دو مثبت بودند ممکن بود جواب خارج از بازه باشد.

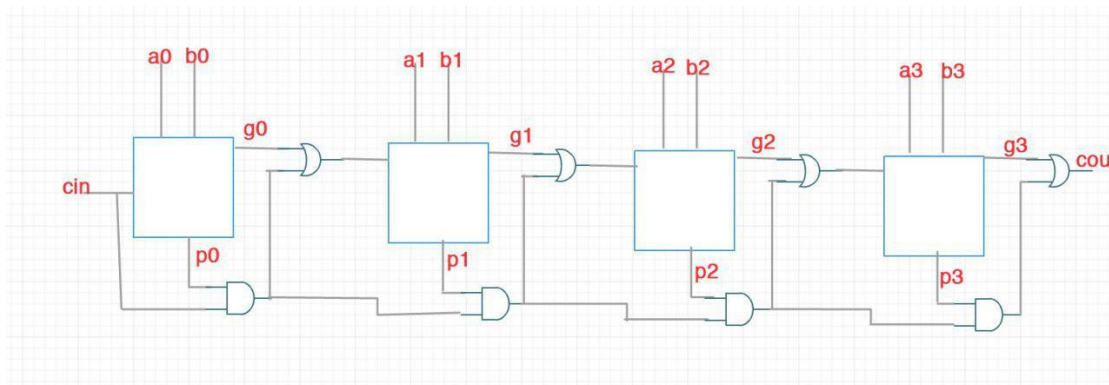
سوال دوم

جمع کننده‌ای بهینه طراحی کنید که

- Single-level carry-skip
- Two-level carry-skip

باشد. (اعداد را ۴ بیتی فرض کنید)

Single-level carry-skip



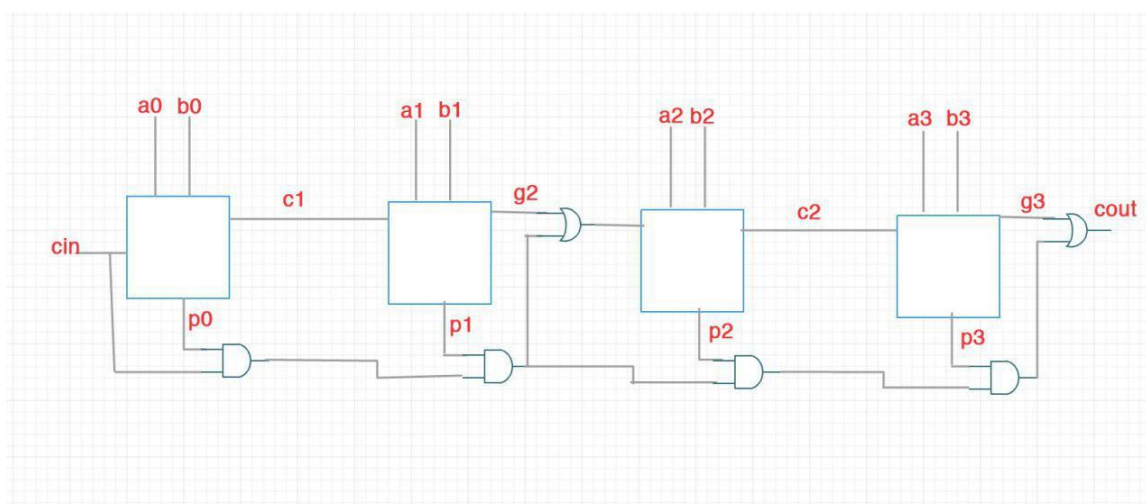
تحويل در روز سه شنبه مورخ ۱۳۹۷/۲/۴

تمرینات خود را با فرمت studentID_studentName_HW4 در مدل آپلود کنید.

لطفا پاسخ ها خوانا و تمیز نوشته شوند.

سوالات و مشکلات خود را به ایمیل computerarchitecture2018@gmail.com بفرستید.

Two-level carry-skip



قسمت الف تنها با استفاده از p و carry قبلی قسمتی از g را پیش محاسبه می کرد که چون با بقیه اعضا در g می شود اگر 1 شود carry بعدی هم 1 شده و نیاز برای انتظار برای محاسبه ی g نیست.

در قسمت ب باید دوبیت دوبیت carry را با استفاده از p and های قبلی و cin (طبق شکل) پیش محاسبه کنیم و ببینیم نیاز برای انتظار برای g هست یا جداگانه محاسبه می شود.

تحويل در روز سه شنبه مورخ ۱۳۹۷/۲/۴

تمرینات خود را با فرمت studentID_studentName_HW4 در مدل آپلود کنید.

لطفا پاسخ ها خوانا و تمیز نوشته شوند.

سوالات و مشکلات خود را به ایمیل computerarchitecture2018@gmail.com بفرستید.

سوال سوم

جمع کننده ۱۵ رقمی دهدهی را برای اعداد ۶۰ بیتی بدون علامت در نظر بگیرید:

- مدار مورد نیاز برای تولید بیت نقلی و زمان انتقال آن را با توجه به باینری بودن اعداد دهدهی، طراحی کنید.
- قسمت بالا را با نحوه excess-3 encoding برای اعداد دهدهی تکرار کنید. (در این حالت مقدار a را به صورت $a + 3$ در مبنای دو نمایش می دهیم)
- طراحی جمع کننده دهدهی قسمت قبل را با قرار دادن مدار carry-lookahead و مدار محاسبه جمع کامل کنید.

اعداد در این جا به صورت BCD نمایش داده می شوند، هر رقم ۴ بیت باینری برای ۱۵ رقم ۶۰ بیت باینری نیاز داریم، شکلی که مشاهده می کنید برای جمع دو رقم است برای طراحی کامل باید ۱۵ تا مدار خواسته شده طراحی گردد و هر جمع کننده بیت C_0 خود را به C_1 جمع کننده ی بعدی بدهد. دقت کنید در BCD اگر overflow داشتیم باید ۴ بیت را با 0110 جمع می کردیم اما این جا برای سهولت فرض شده بلاک BCD Excess-3 Converter این کار را انجام می دهد که اگر رقم از ۹ بزرگتر باشد جمع را انجام می دهد و C_{out} را خروجی می دهد که با طبق شکل OR می شود و C_0 را خروجی می دهد.

در قسمت دوم خواسته شده اعداد Excess-3 باشند که کافی است در مدار BCD Excess-3 Converter ارقام را با 3 به صورت عادی و با استفاده از full adder جمع کنیم و بعد در اختیار جمع



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

معماری کامپیوتر نیمسال دوم ۱۳۹۶ تمرین چهارم



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تحويل در روز سه شنبه مورخ ۱۳۹۷/۲/۴

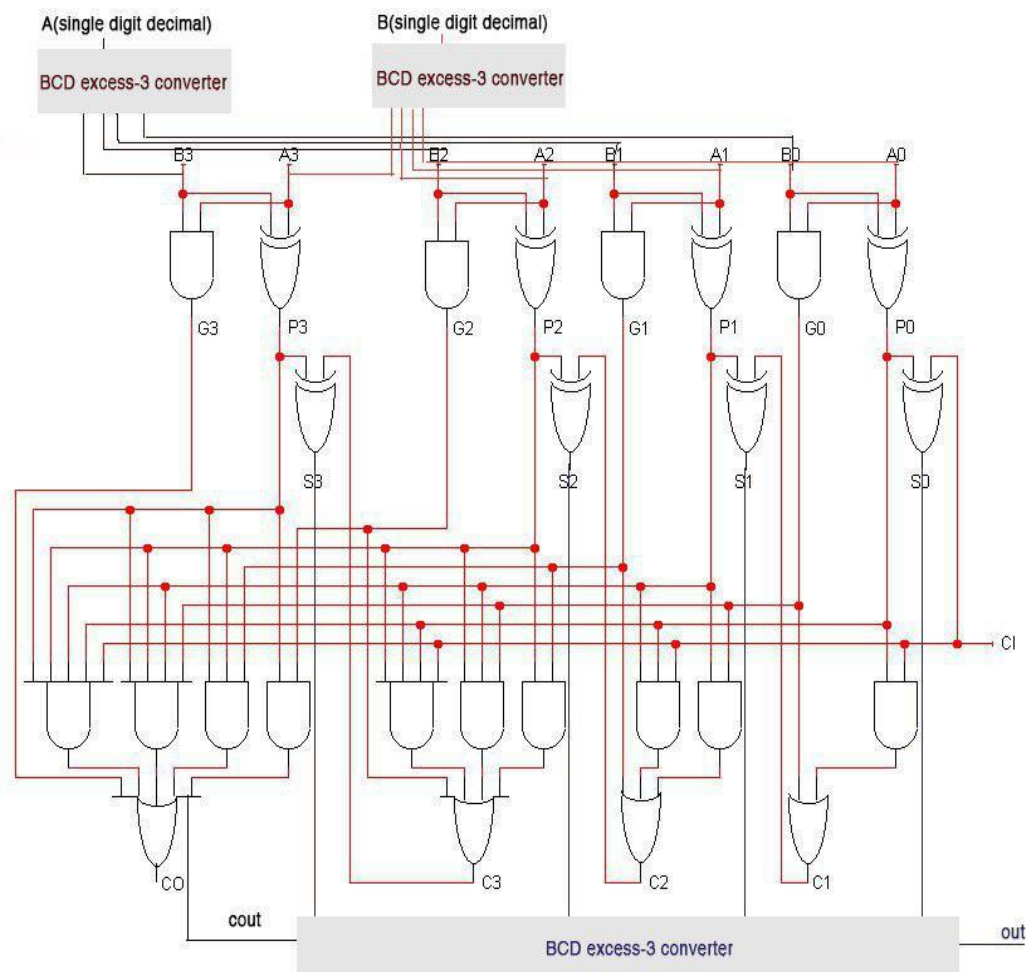
تمرینات خود را با فرمت studentID_studentName_HW4 در مدل آپلود کنید.

لطفا پاسخ ها خوانا و تمیز نوشته شوند.

سوالات و مشکلات خود را به ایمیل computerarchitecture2018@gmail.com بفرستید.

کننده قرار بدهیم. خروجی هم باید Excess-3 باشد که در این صورت تنها کافی است در مدار BCD Excess-3 Converter ابتدا ۳ را از عدد کم کرده و بعد مراحل پاراگراف بالا را روی عدد انجام دهیم.

مدار جمع کننده ی نهایی هم به صورت زیر می باشد:





دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

معماری کامپیوتر
نیمسال دوم ۱۳۹۶
تمرین چهارم



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تحويل در روز سه شنبه مورخ ۱۳۹۷/۲/۴

تمرینات خود را با فرمت studentID_studentName_HW4 در مودل آپلود کنید.

لطفا پاسخ ها خوانا و تمیز نوشته شوند.

سوالات و مشکلات خود را به ایمیل computerarchitecture2018@gmail.com بفرستید.

سوال چهارم

یک نمایش ۶ بیتی ممیز شناور که ۳ بیت آن برای excess-3 و توان استفاده شده است و ۲ بیت آن برای mantissa را در نظر بگیرید.

(۱) نمایش ۶ بیتی 0.75_{10} چگونه است؟

001010_2

ابتدا قسمت اعشاری را از بین می بریم:

$$0.75 * 2 = 1.5 \rightarrow 1$$

$$0.5 * 2 = 1 \rightarrow 1$$

$$0.11 * 2^0 \rightarrow 1.10 * 2^{-1}$$

$$exponent = -1 + 3 = 2 = 010_2$$

$$mantissa = 10$$

$$\rightarrow \text{the number is positive} \rightarrow 0.75_{10} = 001010_2$$

(۲) معادل ده دهی 011010_2 و 000010_2 را بدست آورید.

$$011010_2 = 12_{10}$$

$$000010_2 = 0.1875_{10}$$

تحويل در روز سه شنبه مورخ ۱۳۹۷/۲/۴
تمرینات خود را با فرمت studentID_studentName_HW4 در مودل آپلود کنید.
لطفا پاسخ ها خوانا و تمیز نوشته شوند.
سوالات و مشکلات خود را به ایمیل computerarchitecture2018@gmail.com بفرستید.

برای این سوال راه حل جرئی یک قسمت را توضیح می دهیم:

تبدیل 000010_2 به عدد ده دهی:

0 در MSB بیت علامت است و نشان می دهد عدد مثبت می باشد.

000 در ادامه توان می باشد که excess-3 یعنی

$$0 - 3 = -3 \rightarrow exponent = 2^{-3}$$

10 نیز mantissa را نشان می دهد، عدد کلی برابر

$$1.10 * 2^{-3} \rightarrow 0.00110 * 2^0 = 0.1875$$

سوال پنجم

نمودار زیر بیانگر الگوریتم جمع دو عدد ممیز شناور ۳۲ بیتی است، مراحل خواسته شده در شکل (که با قرمز مشخص شده اند) را توضیح دهید.

Exponent difference: این بلاک اختلاف توان های دو عدد را محاسبه می کند و برای شیفت دهنده مشخص می کند که چند واحد شیفت بدهد تا با یکی شدن توان ها قسمت اعشاری عدد های آن ها قابل جمع شود.

Right Shifter: عدد کوچکتر را شیفت می دهد تا توان ها یکی شوند و قابل جمع شوند.

2's complement adder: دو مقدار اعشاری را که قابل جمع شده اند با هم جمع می کند.



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

معماری کامپیوتر نیمسال دوم ۱۳۹۶ تمرین چهارم



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تحويل در روز سه شنبه مورخ ۱۳۹۷/۲/۴

تمرینات خود را با فرمت studentID_studentName_HW4 در مدل آپلود کنید.

لطفا پاسخ ها خوانا و تمیز نوشته شوند.

سوالات و مشکلات خود را به ایمیل computerarchitecture2018@gmail.com بفرستید.

Normalize: عدد حاصل جمع را دوباره به فرم mantissa در می آوریم و توان اضافی آمده را با اختلاف توان اولیه جمع می کنیم تا توان جمع هم حساب شود.

