(۱

لچ برای نگه داشتن آخرین داده استفاده می شود و به نوعی عملکردی نظیر حافظه را از خود بروز میدهد. از طرفی بافر برای پایدار کردن ولتاژ به کار می رود. معمولا از لچ برای داده خروجی و از بافر برای داده ورودی استفاده می شود. ولی تنها حالتی که از لچ برای پورت ورودی و از بافر در خروجی استفاده کنیم این است که در خروجی پایداری ولتاژ و در ورودی نگه داشتن داده ها برای ما اهمیت داشته باشد.

(۲

پارامترهای زمانی مهم لچ: بعنوان مثال در لچ 82C82 پارامتر های زمانی مهم بدین ترتیب هستند(در لچ های دیگر نیز پارامترهای مشابه و معادل همین پارامترها وجود دارند)

- tSHSL: زمان high بودن پایه STB(Strobe)
- tIVSL: فاصله بین زمان گذاشتن داده تا لبه پایین رونده سیگنال STB
- tSLIX: فاصله بین زمان پایین رفتن پالس STB تا انتهای زمان معتبر بودن داده

پارامترهای زمانی مهم بافر: بعنوان مثال در بافر 74LS244 پارامترهای زمانی مهم بدین ترتیب هستند(در بافرهای دیگر نیز پارامترهای مشابه و معادل همین پارامترها وجود دارند)

- tPLH: تاخیر انتشار[1] برای تغییر وضعیت خروجی از Low به High با فرض فعال بودن پایه های enable
- tPHL: تاخیر انتشار برای تغییر وضعیت خروجی از High به Low با همان فرض
- tPZL: تاخیر انتشار برای رفتن خروجی از حالت شناور به وضعیت Low
- tPZH: تاخیر انتشار برای رفتن خروجی از حالت شناور به وضعیت High

پارامترهای زمانی مهم دیکودر: بعنوان مثال در دیکودر 74LS138 پارامترهای زمانی مهم بدین ترتیب هستند(در بافرهای دیگر نیز پارامترهای مشابه و معادل همین پارامترها وجود دارند)

- tPLH: تاخیر انتشار از زمان گذاشتن آدرس تا تغییر وضعیت خروجی از High به Low با فرض فعال بودن پایه های enable(علت تفاوت تعریف tPLH در دیکودر با tPLH در بافر این است که پایه های خروجی دیکودر active low میباشند)
- tPHL: تاخیر انتشار از زمان گذاشتن آدرس تا تغییر وضعیت خروجی از Low به High با فرض فعال بودن پایه های enable

پارامترهای زمانی مهم انکودر: بعنوان مثال در انکودر 74148 پارامترهای زمانی مهم دقیقا مثل پارامترهای زمانی در دیکودر میباشند(در انکودرهای دیگر نیز پارامترهای مشابه و معادل همین پارامترها وجود دارند)

(۳

روش سرکشی علی رغم پیاده سازی سخت افزاری و نرم افزاری ساده، روش مناسبی برای ارتباط وسایل جانبی و پردازنده نیست. زیرا پردازنده مدام باید سیگنال خاصی را چک کند تا به وضعیت مطلوب (مثلا برای سیگنال Busy مقدار 0 وضعیت مطلوب را نشان میدهد) برسد و این کار باعث کاهش راندمان می شود. زمانیکه چند دستگاه جانبی وجود دارد روش سرکشی ممکن است کاملا نارضایت بخش باشد. برای همین معمولا از روش وقفه استفاده می کنند.

[1] Propagation Delay

در روش های مبتنی بر وقفه، چک کردن سیگنال دیگر بطور نرم افزاری بررسی نمی شود و با رسیدن سیگنال دستگاه های جانبی به سطح مطلوب، این سیگنال ها به طور سخت افزاری وقفه ای تولید میکنند و باعث میشوند اجرای برنامه جاری پردازنده متوقف شده و به روتین وقفه منتقل گردد. در این روش پردازنده در مدت زمانی که آمادگی دستگاه های جانبی توسط سیگنال های وقفه اعلام نشده میتواند به کارهای دیگر خود مشغول باشد و بدین ترتیب راندمان نسبت به روش سرکشی تا حدودی بهتر میشود.

۴)

در حالت مبتنی بر وقفه، متداول ترین کار استفاده از Priority Encoder است. در این حالت چنانچه دو تقاضای وقفه همزمان صورت گیرد، encoder وسیله با اولویت بیشتر را کدگذاری و شماره کدگذاری شده را در خروجی قرار میدهد.

در حالت سرکشی، میتوان با قائل شدن نوعی نسبت و وزن بین تعداد دفعات سرکشی برای دستگاه های مختلف، این اولویت بندی را رعایت کرد. مثلا به ازای هر ۳ بار سرکشی برای دستگاه با اولویت بالاتر، صرفا یک بار دستگاه با اولویت پایین تر را نیز سرکشی کنیم.

۵)

در سئوال ۴ روش انجام این کار ذکر شده است.

۶)

در ریزپردازنده Atmega16، مقدار رجیستر PC برای ذخیره سازی آدرس بازگشت هنگام انتقال کنترل به ISR در پشته قرار میگیرد و هنگام بازگشت به پردازش عادی و اتمام ISR، آدرس بازگشت نهایتا از پشته خونده میشود. ولی ممکن است علاوه بر آدرس بازگشت، اطلاعاتی نظیر مقادیر برخی رجیسترهای خاص (از مهم ترین آنها میتوان به SREG حاوی وضعیت پرچم های پردازنده اشاره کرد) حائز اهمیت باشد که در این صورت باید آن ها را نیز در ابتدای ISR با دستور push در پشته قرار داده و در انتهای ISR با دستور pop از پشته خوانده و مقادیر قبل از ISR شان را بازیابی کنیم تا اجرای برنامه اصلی و پردازش عادی پس از ISR به مشکل نخورد.

۷)

الف)

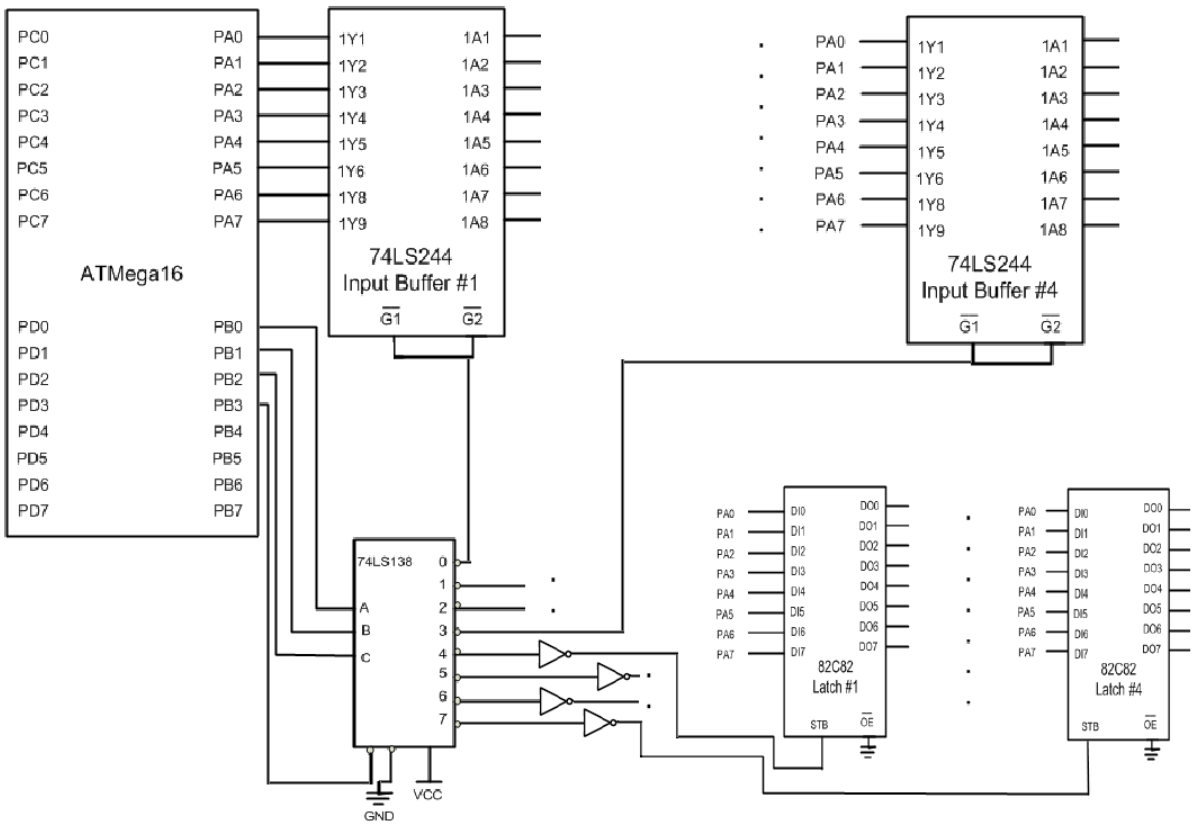
از 8 خروجی دیکودر 74LS138, 4 تای اول به بافر 74LS244 و 4 تای بعدی به لچ 82C82 اختصاص داده شده اند. پایه Strobe لچ ها برخلاف Enable برای Input Buffer ها که active low هستند، active high می باشد. برای همین در مسیر آن ۴ خروجی که از دیکودر به لچ 82C82 وصل شده، گیت NOT قرار گرفته است.

در حالتی که نه به بافر ها نیاز داریم و نه به لچ ها، پایه E1 یا همان G2A از دیکودر در وضعیت high قرار میگیرد تا طبق جدول درستی دیکودر، تمامی خروجی های دیوکدر high شوند و در نتیجه هیچ دستگاهی انتخاب نشود.

در غیر اینصورت، به هر دستگاهی که نیاز داشته باشیم کافیست تا ابتدا پایه G2A از دیکودر را در وضعیت low قرار دهیم سپس شماره خروجی مورد نیاز برای کار با دستگاه مد نظر را در به ورودی SELECT از دیکودر بدهیم و در نهایت نیز با رعایت پارامترهای زمانی مربوط، کارمان را با آن دستگاه انجام دهیم. در انتهای کار نیز پایه G2A از دیکودر را مجددا در وضعیت high قرار می دهیم.

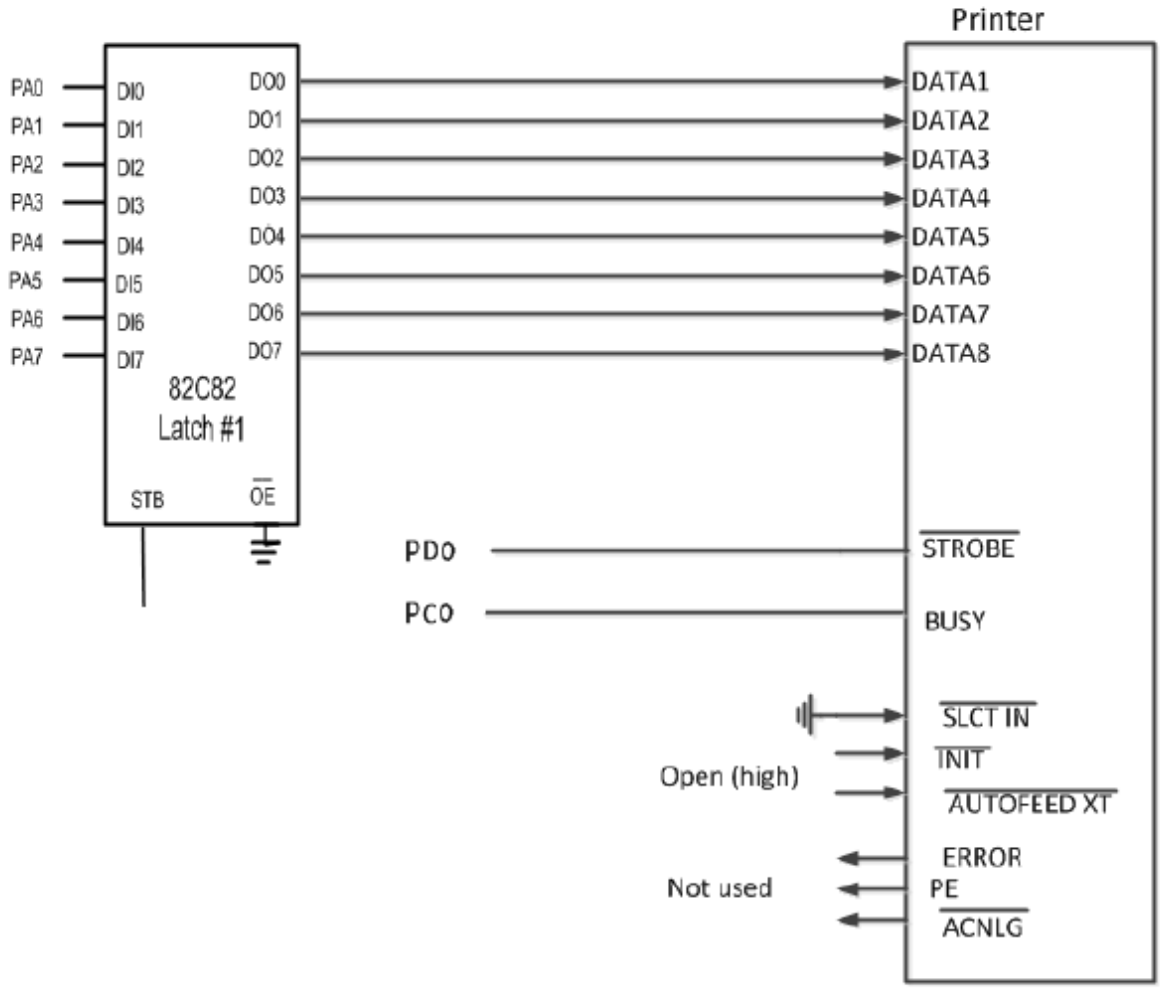
لازم به ذکر است در این جا پایه G2A از دیکودر، به زمین وصل نیست و به پایه PB3 از پورت B میکروکنترلر متصل میباشد.

شماتیک مدار به شرح زیر است:

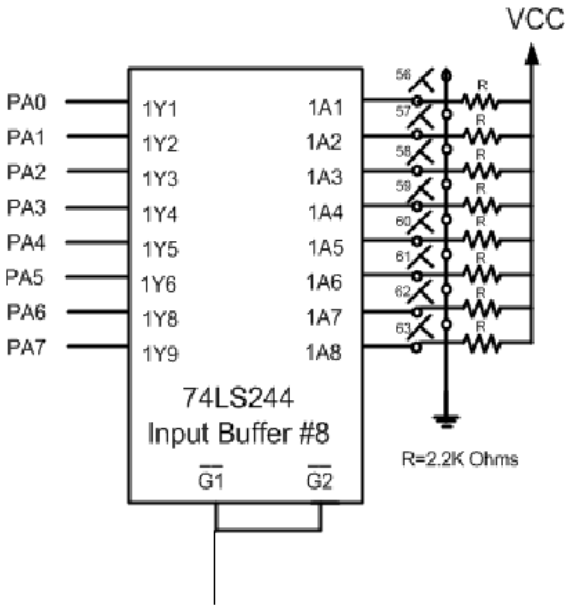


ب)

برای یکی از درگاه های خروجی نحوه بستن کلید ها بدین ترتیب میباشد:



برای یکی از درگاه های ورودی نیز نحوه اتصال به پرینتر بدین ترتیب میباشد:



نحوه اتصال برای سایر درگاه های ورودی و یا خروجی مشابه است.

کد خواندن کلید فشار داده ورودی به شرح زیر است. در اینجا فرض شده تنظیمات ورودی و یا خروجی پورت ها قبلا انجام گرفته است.

```
CALL BufferRead

BufferRead:

RCALL Poll_Buff1

RCALL Poll_Buff2

RCALL Poll_Buff3

RCALL Poll_Buff4

RJMP BufferRead

Poll_Buff1:

LDI Index, 0;for Poll_Buff2,3,4 Index should be loaded with 8, 16 and 24 respectively

LDI R20, 8

LDI R24, 0x00;To poll buffer #1. For Poll_buffer2,3,4 the immediate value should be 0x01, 0x02 and 0x03
;respectively

OUT PORTB, R24

CBI PORTB, PB3;Decoder’s outputs would be determined based on SELECT lines

NOP

NOP
```

```
NOP;3 * NOP = 3 * 62.5 > 41ns(tPHL, decoder) + 30ns(tPZL, Buffer) + 1.5 * 62.5ns(PIN Register)

IN R16, PINA

CMP R16, 0XFF


BREQ Poll_Buff1_Terminated

RCALL Delay20ms;Call a 20ms Delay if any key was pressed. In order for possible noise to be
;compensated

Loop_Buff1:

DEC R20

LSL R16

BRCC Poll_Buff1_Terminated

RJMP Loop_Buff1

Poll_Buff1_Terminated:

SBI PORTB, PB3;All of the decoder's outputs should be set to 1

ADD R20, Index;Now, R20 containts the number of the pressed key

RET

;Poll_Buff2,3,4 can be implemented just as Poll_Buff1. There isn't a noticeable difference. All needed is
;to replace LDI R16, PINA with a corresponding value stated in comments. Also replace each Buff1 with
;Buff2
```

کد ارسال کلید فشار داده شده به خروجی برای پرینتر به شرح زیر است. در اینجا فرض شده تنظیمات ورودی و یا خروجی پورت ها قبلا انجام گرفته است. مقدار مورد نیاز در رجیستر R20 قرار دارد که با صدا زدن زیرروال Convert_ASCII به معادل ASCII برای چاپ کردن در پرینتر تبدیل شده است.

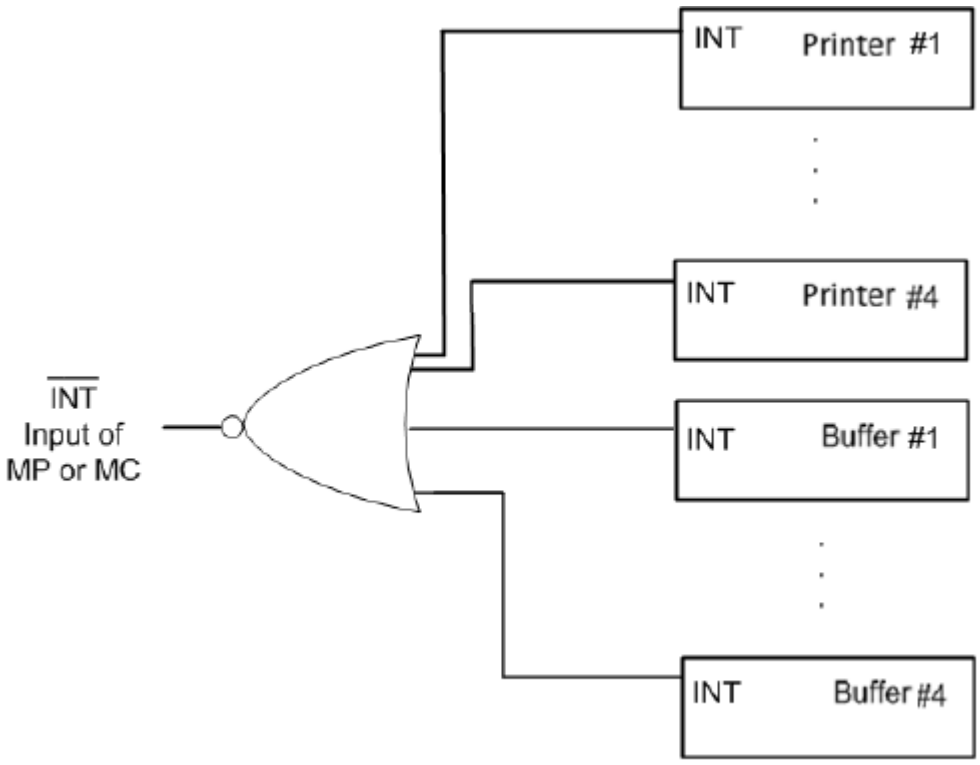
```
RCALL Printer1;For printing the data concerning to Buffer1

Printer1:

SBIC PINC, PC0;Busy line for Printer1. It should be replaced by PC1, PC2 and PC3 for Printer2, Printer3
;and Printer4 respectively

RJMP Printer1

OUT PORTA, R20;Value on PORTA;(tIVSL = 0ns satisfied)

LDI R16, 0x04;The immediate value for Printer2,3,4 should be replaced by 0x05, 0x06 and 0x07
;respectively
```

```
OUT PORTB, R16

CBI PORTB, PB3;Latch#1 Strobe high

NOP

NOP;2 * NOP = 2 * 62.5 = 125  > 41ns(tPHL, decoder) + 35ns(tSHSL)

SBI PORTB, PB3;Latch#1 Strobe low and set all of the decoder’s outputs to 1

NOP;1 * NOP = 1 * 62.5 > tSLIX = 25ns

CBI PORTD, PD0;Printer Strobe line for Printer1. It should be replaced by PD1, PD2 and PD3 for Printer2,
;Printer3 and Printer4 respectvively

NOP

NOP

NOP

NOP

NOP

NOP

NOP

NOP;Strobe should be logical low for duration of 500ns = 8 * 62.5ns = 8 * NOP

SBI PORTD, PD0;Printer Strobe = 1

RET
```

ج)

برای این که رخداد وقفه توسط منبع وقفه، برای پایه وقفه پردازنده، یک سطح پایین یا همان صفر منطقی تولید کند باید پایه وقفه پردازنده را به نوعی active low قرار دهیم.

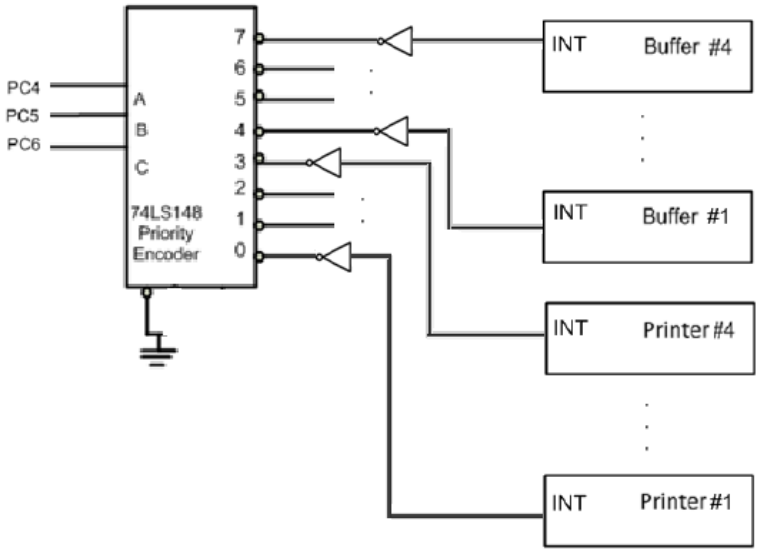
در حالتی که منابع وقفه، پایه های وقفه شان activce high و پایه وقفه میکروکنترلر active low بوده باشد باید از گیت NOR مطابق شکل زیر استفاده کنیم:



ضمنا برای تشخیص کدام وسیله، برای این که قسمت بعد را هم رعایت کرده باشیم، از انکودر 74LS148 استفاده کرده ایم که یک priority encoder است. لازم به ذکر است بعلت active low بودن تمامی پایه های این انکودر، از گیت NOT بین ورودی های انکودر و پایه های وقفه منابع وقفه استفاده کرده ایم.

هر چه شماره منبعی که وقفه low میدهد بالاتر باشد(دقت کنید با گیت های not وقفه ها در ورودی دیکودر low می شوند) اولویت آن منبع بیشتر است. برای همین ورودی شماره 7 تا 4 را به صفحه کلیدها و ورودی های شماره 3 تا 0 را به پرینتر ها اختصاص داده ایم.

خروجی encoder به ورودی های PC4,5,6 از پورت C متصل است. ورودی های 0 الی 3 از پورت C در سئوالات قبل برای کارهای دیگر استفاده شده اند.



د)

INT0_ISR:

NOP

NOP;for preserving the 1.5 clock limitation of PINC

IN R16, PINC

LSR R16

LSR R16

LSR R16

LSR R16;Now R16 contains the output of encoder containint the number of interrupting device ready to
;be serviced, in active low format

COM R16;The correct number of interrupting device in active high format

ANDI R16, Ob00000111

CPI R16, 7

BREQ Poll_Buff4_Call

```
CPI R16, 6
BREQ Poll_Buff3_Call
CPI R16, 5
BREQ Poll_Buff2_Call
CPI R16, 4
BREQ Poll_Buff1_Call
CPI R16, 3
BREQ Printer4_Call
CPI R16, 2
BREQ Printer3_Call
CPI R16, 1
BREQ Printer2_Call
CPI R16, 0
BREQ Printer1_Call
AFTER:
RETI

Printer1_Call:
CALL Printer1
RJMP AFTER

Printer2_Call:
CALL Printer2
RJMP AFTER

Printer3_Call:
CALL Printer3
RJMP AFTER
```

```
Printer4_Call:
CALL Printer4
RJMP AFTER

Poll_Buff1_Call:
CALL Poll_Buff1
RJMP AFTER

Poll_Buff2_Call:
CALL Poll_Buff2
RJMP AFTER

Poll_Buff3_Call:
CALL Poll_Buff3
RJMP AFTER

Poll_Buff4_Call:
CALL Poll_Buff4
RJMP AFTER
```