

Architecture des ordinateurs 2

Cours 6 - Instructions Assembleur pour AVR Atmega 328p

Halim Djerroud <hdd@ai.univ-paris8.fr>

LIASD - Université Paris 8

Janvier 2018

Vider un registre

- Vider un registre
- $Rx \oplus Rx$
- Instruction **CLR**

```
...  
clr R15  
...
```

Incrémenter / Décrémenter

- $INC \Rightarrow +1$
- $DEC \Rightarrow -1$
- Instructions **INC**, **DEC**

```
...  
INC R15  
DEC R16  
...
```

Mettre tout à 1

- Mettre tout les bits à 1
- SER => set to 0xFF

```
...  
SER R15  
...
```

Charger une valeur constante

- LDI => Load Immediate

```
...  
LDI R15, 10           ; mettre 10 dans R15  
LDI R12, 0b00000010  ; mettre 2  dans R12  
LDI R12, 0x1A        ; mettre 32 dans R12  
LDI R12, 010         ; mettre 8  dans R12  
LDI R13, '1'         ; mettre 49 dans R13  
...
```

Copier un registre

- MOV => Move Between Registers
- Copier le contenu d'un registre dans un autre registre

```
...  
MOV R24, R12  
...
```

Addition sans retenue

- Addition sans retenue
- ADD

...

```
LDI R24, 10
```

```
LDI R25, 100
```

```
ADD R24, R25
```

...

Addition avec retenue

- Addition avec retenue
- ADC

```
...  
LDI R16, lo8(1650)  
LDI R17, hi8(1650)  
LDI R18, lo8(5250)  
LDI R19, hi8(5250)  
ADD R16, R18  
ADC R17, R19  
...
```


Autres Opérateurs

- Soustraction valeur immédiate **SUBI** sans carry
- Soustraction valeur immédiate **SBCI** sans carry
- Soustraction deux registres **SUB**
- Soustraction deux registres **SBC** avec carry
- MUL

if else

```
...  
ldi r16, 5  
ldi r17, 5  
cp r16, r17  
breq // si égale =  
    ... // code non exécuté  
jmp end  
else:  
    ... // code exécuté  
end:  
...
```

if else

```
...  
ldi r16, 5  
ldi r17, 7  
cp r16, r17  
brlo // <  
...  
jmp end  
else:  
...  
end:  
...
```

if else

```
...  
ldi r16, 5  
ldi r17, 7  
cp r16, r17  
brsh // >=  
...  
    jmp end  
else:  
    ...  
end:  
    ...
```

if else

```
...  
ldi r16, 5  
ldi r17, 7  
cp r16, r17  
brlt // <  
...  
jmp end  
else:  
...  
end:  
...
```

if else

```
...  
ldi r16, 1  
ldi r17, 1  
cpse r16, r17  
rcall ailleurs  
... //
```

for(;;)

```
    ...  
end:  
    ...  
    rjmp end
```

Exemple boucle infinie

```
main:  
    ...  
    rjmp main  
  
shutdown:  
    ...  
  
end:  
    rjmp end
```


for

```
    ldi r16, 10  
for:  
    ...  
  
    dec r16  
    brne for  
    ...
```

for généré par le code C

```
/* for(i = 10; i > 0; i--) */  
  
ldi R16, 10  
clz  
for:  
    breq end  
    push R16 // sauvgarder  
    ...  
    pop r16 // Réstaurer  
    dec R16  
    rjmp for  
end:  
    ...
```

do-while

```
    ldi R18, 15
loop:
    ...
    subi R18, 1
    cpi  R18, 0
    brcc loop
```

while

```
    ldi R18, 15
while:
    subi R18, 1    // autre choses
    cpi  R18, 0
    brcc while

    ...

    rjmp while
```

Fréquence

- Oscillateur Arduino externe 16 Mhz
- Oscillateurs possibles en interne 4Mh et 8Mhz
- Calculer le délai que dure 1 Cycle d'horloge pour chaque fréquence

Signal

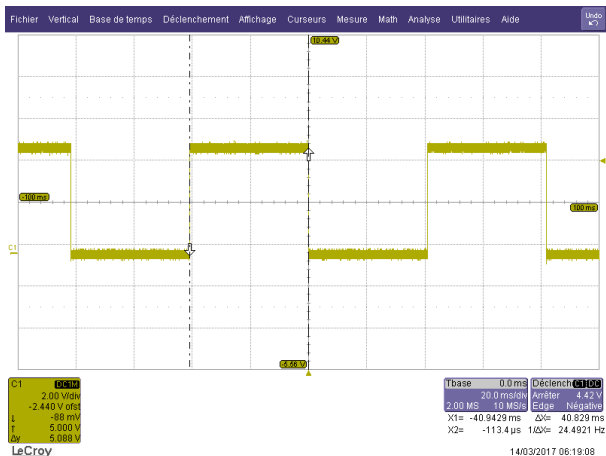


FIGURE – Capture oscilloscope pour 1 Instruction Atmega 328p -16Mhz

Delay

```
    ...  
    ldi R18, 255  
    ldi R19, 255  
    ldi R20, 10  
    rcall delay  
    ...  
delay:  
    subi R18, 1  
    sbci R19, 0  
    sbci R20, 0  
    brcc delay  
    ret
```