

```
#####
#          cs315 Week 12 - part (B)
#
#  -> Hand assembly, jump address calculation
#
#####
```

- address -	- instruction -	- machine code -	- notes -
0x0040 81AC	jal r_in	0000 11ii iiiiiiii iiiiiiii	jal 0x0040 B6CC
...		0000 1100 0001 0000 0010 1101 1011 0011	0x0040 B6CC = 0000 0000 0100 0000 1011 0110 1100 1100 =
...			^
...			1) remove first nibble 2) remove last 2 bits (word boundary)
...			
0x0040 B6CC	r_in: addi \$9, \$9, \$0		--> 0000 0100 0000 1011 0110 1100 11 <-- jump target address

Notes:

1) remove the first nibble because addresses in .text (or assembly codes) section are in the following format:

```
0x0____
  ^
```

common in all addresses in .text, so no need to store

2) remove last 2 bits because addresses are in word boundary, or divisible by 4

HEX:

```
0 --> 0000
4 --> 0100
8 --> 1000
C --> 1100
      ^^
```

zeros in the last 2 bits, so no need to store