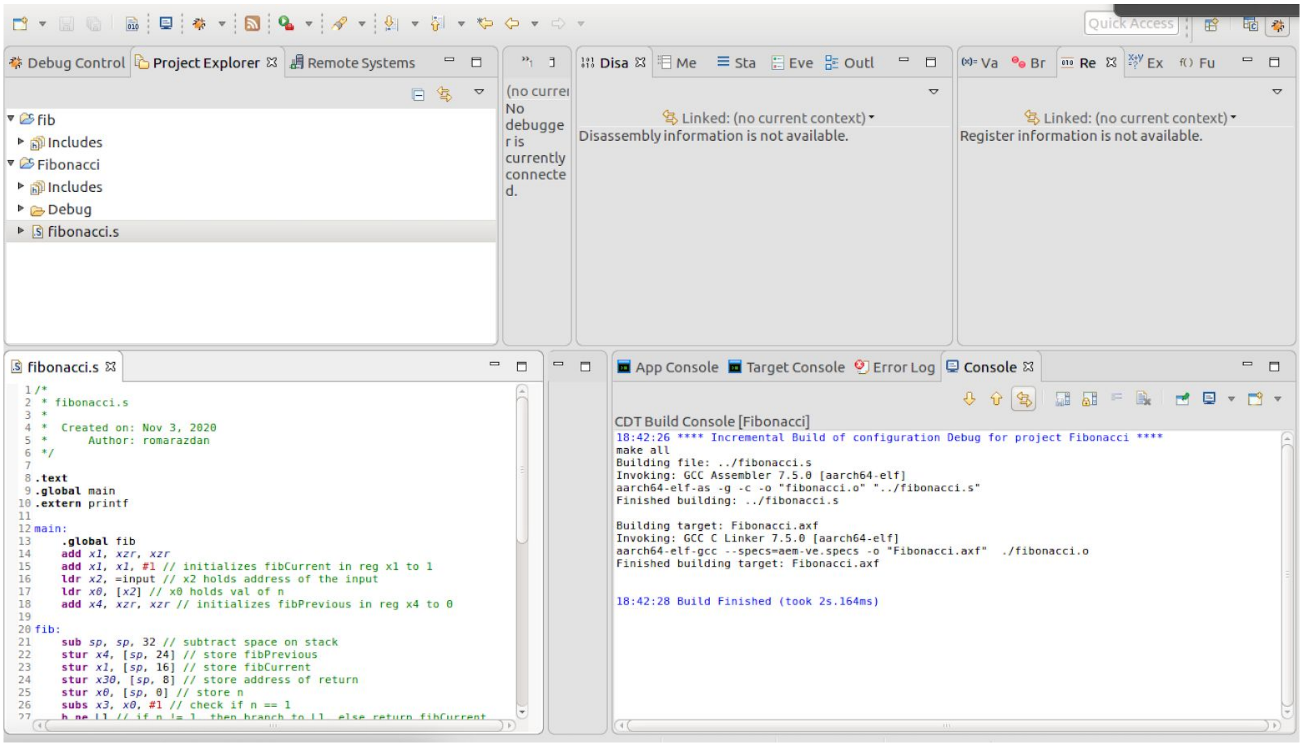
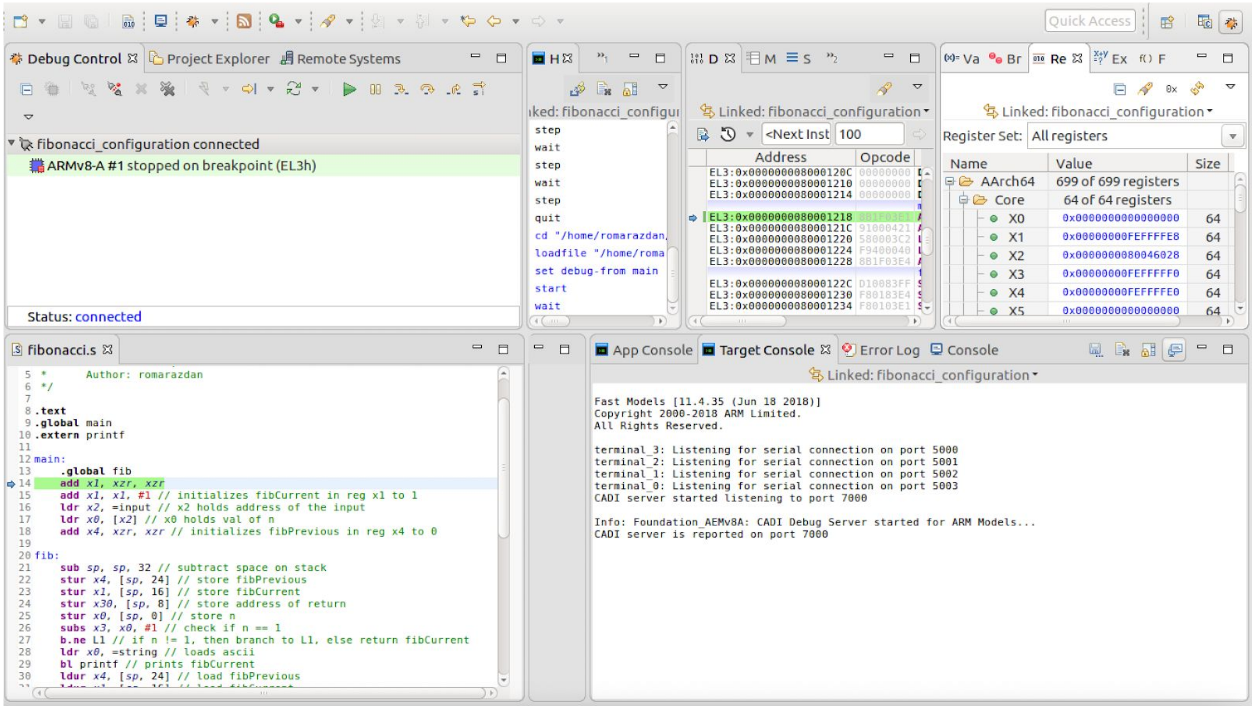
Using the example code of recursive fibonacci I was able to produce the assembly code of it over the past few days. One of my biggest problems at first was determining how to set up and use stack in assembly language. After figuring out how to utilize the stack feature it made my life a whole lot easier as I was able to subtract, add space off and on the stack very easily. When using stack in assembly we must consider that stack has a last in first out ordering principle (LIFO) thus the registers that are used must be in order of x0-x5, x9-x13, and x30.

The process of my code follows the following logic: First I must initialize the fibCurrent as well as fibPrevious in two separate registers, then I must hold the address of the input data, and value of n. Following those steps in my main, now I can move onto the fib where I store all of these values. Then I must check if n == 1, and if n != 1, then branch to loop one (which I will address in the following sentences), else return fibCurrent. Then I load ascii representation of string, then print out the value of fibCurrent. Next, I must load these values and add space back onto the stack. Finally return to the caller. In the loop one I mentioned earlier I follow a similar logic, however this time I might decrement n since it did not fit the criteria of n == 1. Eventually I go through similar steps and return to caller the value of fibCurrent. Finally in the last few lines of my code I input the data and add the print statement!!


Quick Access
Debug Control
Project Explorer
Remote Systems
fib
Includes
Fibonacci
Includes
Debug
fibonacci.s
(no current
No debugger is currently connected.
Disa
Me
Sta
Eve
Outl
Linked: (no current context)
Disassembly information is not available.
Va
Br
Re
Ex
Fu
Linked: (no current context)
Register information is not available.
fibonacci.s
App Console
Target Console
Error Log
Console
CDT Build Console [Fibonacci]

```
/*
 * fibonacci.s
 *
 *  Created on: Nov 3, 2020
 *      Author: romarazdan
 */

.text
.global main
.extern printf

main:
    .global fib
    add x1, xzr, xzr
    add x1, x1, #1 // initializes fibCurrent in reg x1 to 1
    ldr x2, =input // x2 holds address of the input
    ldr x0, [x2] // x0 holds val of n
    add x4, xzr, xzr // initializes fibPrevious in reg x4 to 0

fib:
    sub sp, sp, 32 // subtract space on stack
    stur x4, [sp, 24] // store fibPrevious
    stur x1, [sp, 16] // store fibCurrent
    stur x30, [sp, 8] // store address of return
    stur x0, [sp, 0] // store n
    subs x3, x0, #1 // check if n == 1
    b.ne L1 // if n != 1  then branch to L1  else return fibCurrent
```

```
18:42:26 **** Incremental Build of configuration Debug for project Fibonacci ****
make all
Building file: ../fibonacci.s
Invoking: GCC Assembler 7.5.0 [aarch64-elf]
aarch64-elf-as -g -c -o "fibonacci.o" "../fibonacci.s"
Finished building: ../fibonacci.s

Building target: Fibonacci.axf
Invoking: GCC C Linker 7.5.0 [aarch64-elf]
aarch64-elf-gcc --specs=aem-ve.specs -o "Fibonacci.axf"  ./fibonacci.o
Finished building target: Fibonacci.axf


18:42:28 Build Finished (took 2s.164ms)
```



fibonacci_configuration connected
ARMv8-A #1 stopped on breakpoint (EL3h)

Status: connected

```
ked: fibonacci_configu
step
wait
step
wait
step
quit
cd "/home/romarazdan
loadfile "/home/roma
set debug-from main
start
wait
```

Linked: fibonacci_configuration

<Next Inst 100

```
Address                     Opcode
EL3:0x000000008000120C      00000000
EL3:0x0000000080001210      00000000
EL3:0x0000000080001214      00000000

EL3:0x0000000080001218      8B1F03E1
EL3:0x000000008000121C      91000421
EL3:0x0000000080001220      580003C2
EL3:0x0000000080001224      F9400040
EL3:0x0000000080001228      8B1F03E4

EL3:0x000000008000122C      D10083FF
EL3:0x0000000080001230      F80183E4
EL3:0x0000000080001234      F80103E1
```

Linked: fibonacci_configuration

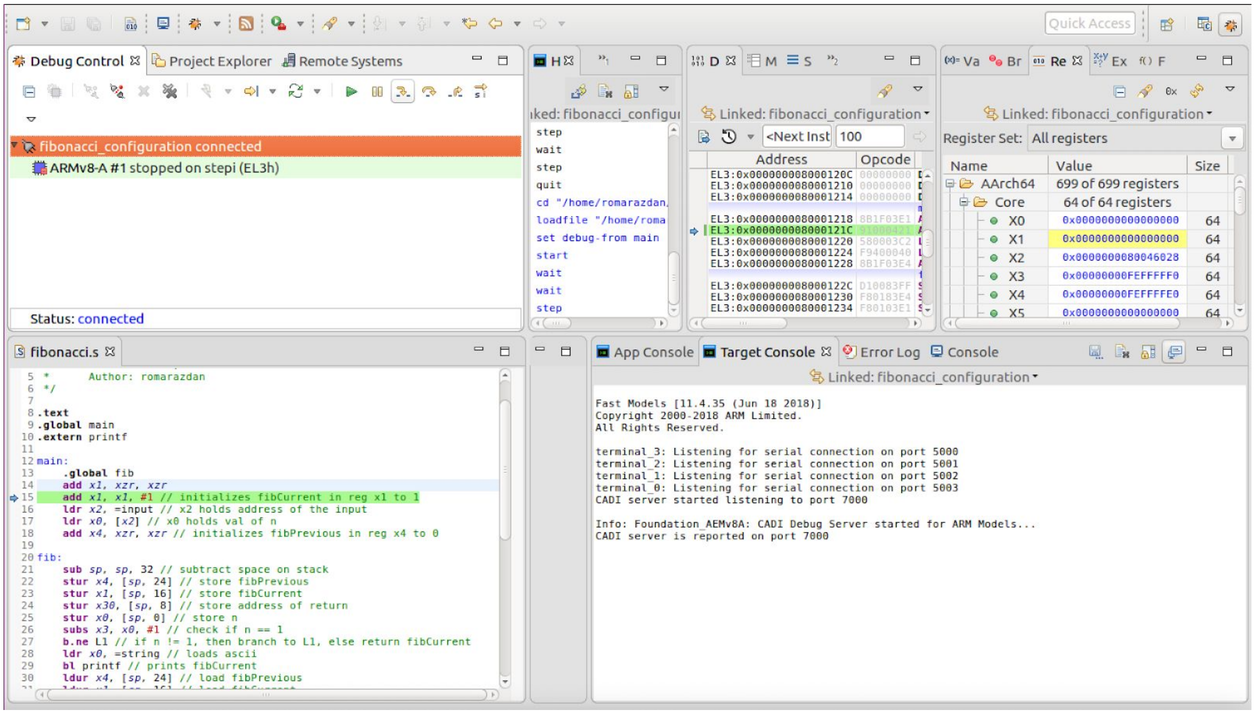
Register Set: All registers

Name	Value	Size
AArch64	699 of 699 registers	
Core	64 of 64 registers	
X0	0x0000000000000000	64
X1	0x00000000FEFFFFE8	64
X2	0x0000000080046028	64
X3	0x00000000FEFFFFF0	64
X4	0x00000000FEFFFFE0	64
X5	0x0000000000000000	64

```
 *      Author: romarazdan
 */

.text
.global main
.extern printf

main:
    .global fib
    add x1, xzr, xzr
    add x1, x1, #1 // initializes fibCurrent in reg x1 to 1
    ldr x2, =input // x2 holds address of the input
    ldr x0, [x2] // x0 holds val of n
    add x4, xzr, xzr // initializes fibPrevious in reg x4 to 0

fib:
    sub sp, sp, 32 // subtract space on stack
    stur x4, [sp, 24] // store fibPrevious
    stur x1, [sp, 16] // store fibCurrent
    stur x30, [sp, 8] // store address of return
    stur x0, [sp, 0] // store n
    subs x3, x0, #1 // check if n == 1
    b.ne L1 // if n != 1, then branch to L1, else return fibCurrent
    ldr x0, =string // loads ascii
    bl printf // prints fibCurrent
    ldur x4, [sp, 24] // load fibPrevious
```

Linked: fibonacci_configuration

```
Fast Models [11.4.35 (Jun 18 2018)]
Copyright 2000-2018 ARM Limited.
All Rights Reserved.

terminal_3: Listening for serial connection on port 5000
terminal_2: Listening for serial connection on port 5001
terminal_1: Listening for serial connection on port 5002
terminal_0: Listening for serial connection on port 5003
CADI server started listening to port 7000

Info: Foundation_AEMv8A: CADI Debug Server started for ARM Models...
CADI server is reported on port 7000
```



fibonacci_configuration connected
ARMv8-A #1 stopped on stepi (EL3h)

Status: connected

```
ked: fibonacci_configu
step
wait
step
quit
cd "/home/romarazdan
loadfile "/home/roma
set debug-from main
start
wait
wait
step
```

Linked: fibonacci_configuration

<Next Inst 100

```
Address                     Opcode
EL3:0x000000008000120C      00000000
EL3:0x0000000080001210      00000000
EL3:0x0000000080001214      00000000

EL3:0x0000000080001218      8B1F03E1
EL3:0x000000008000121C      91000421
EL3:0x0000000080001220      580003C2
EL3:0x0000000080001224      F9400040
EL3:0x0000000080001228      8B1F03E4

EL3:0x000000008000122C      D10083FF
EL3:0x0000000080001230      F80183E4
EL3:0x0000000080001234      F80103E1
```

Linked: fibonacci_configuration

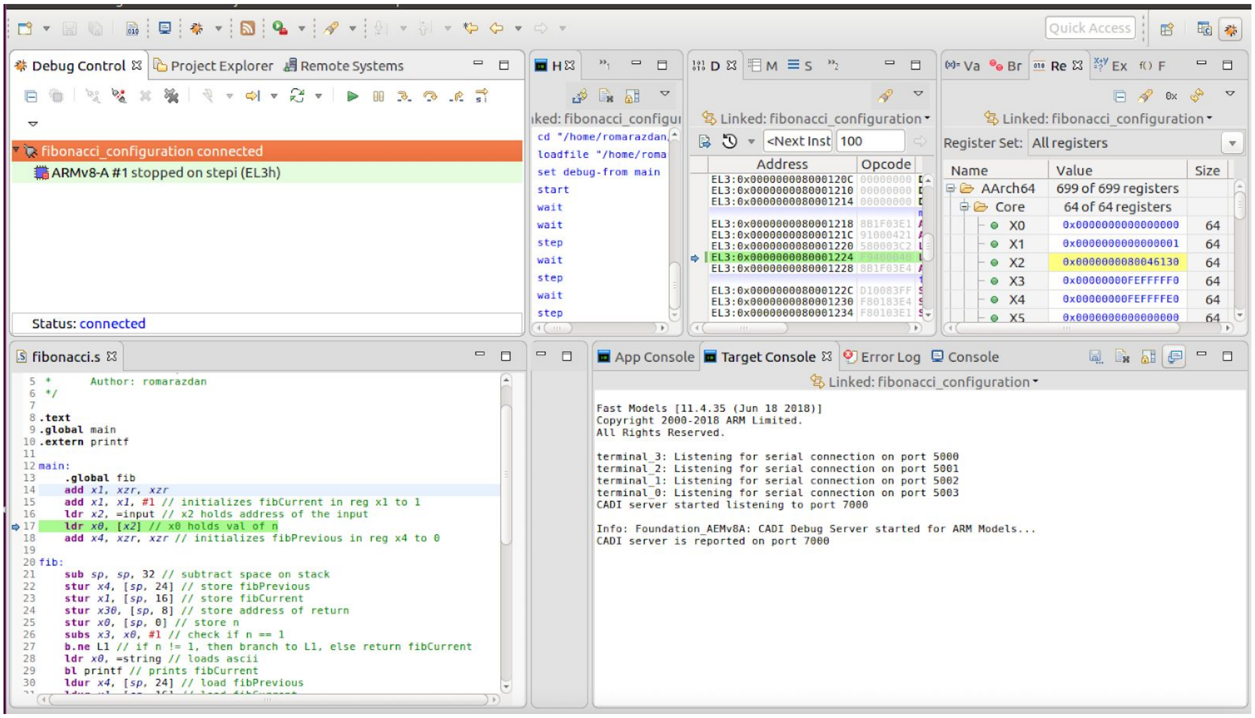
Register Set: All registers

Name	Value	Size
AArch64	699 of 699 registers	
Core	64 of 64 registers	
X0	0x0000000000000000	64
X1	0x0000000000000000	64
X2	0x0000000080046028	64
X3	0x00000000FEFFFFF0	64
X4	0x00000000FEFFFFE0	64
X5	0x0000000000000000	64

```
 *      Author: romarazdan
 */

.text
.global main
.extern printf

main:
    .global fib
    add x1, xzr, xzr
    add x1, x1, #1 // initializes fibCurrent in reg x1 to 1
    ldr x2, =input // x2 holds address of the input
    ldr x0, [x2] // x0 holds val of n
    add x4, xzr, xzr // initializes fibPrevious in reg x4 to 0

fib:
    sub sp, sp, 32 // subtract space on stack
    stur x4, [sp, 24] // store fibPrevious
    stur x1, [sp, 16] // store fibCurrent
    stur x30, [sp, 8] // store address of return
    stur x0, [sp, 0] // store n
    subs x3, x0, #1 // check if n == 1
    b.ne L1 // if n != 1, then branch to L1, else return fibCurrent
    ldr x0, =string // loads ascii
    bl printf // prints fibCurrent
    ldur x4, [sp, 24] // load fibPrevious
```

Linked: fibonacci_configuration

```
Fast Models [11.4.35 (Jun 18 2018)]
Copyright 2000-2018 ARM Limited.
All Rights Reserved.

terminal_3: Listening for serial connection on port 5000
terminal_2: Listening for serial connection on port 5001
terminal_1: Listening for serial connection on port 5002
terminal_0: Listening for serial connection on port 5003
CADI server started listening to port 7000

Info: Foundation_AEMv8A: CADI Debug Server started for ARM Models...
CADI server is reported on port 7000
```



fibonacci_configuration connected
ARMv8-A #1 stopped on stepi (EL3h)

Status: connected

```
ıked: fibonacci_configuı
cd "/home/romarazdan,
loadfile "/home/roma
set debug-from main
start
wait
wait
step
wait
step
wait
step
```

Linked: fibonacci_configuration

<Next Inst 100

```
Address                     Opcode
EL3:0x000000008000120C      00000000
EL3:0x0000000080001210      00000000
EL3:0x0000000080001214      00000000

EL3:0x0000000080001218      8B1F03E1
EL3:0x000000008000121C      91000421
EL3:0x0000000080001220      580003C2
EL3:0x0000000080001224      F9400040
EL3:0x0000000080001228      8B1F03E4

EL3:0x000000008000122C      D10083FF
EL3:0x0000000080001230      F80183E4
EL3:0x0000000080001234      F80103E1
```

Linked: fibonacci_configuration

Register Set: All registers

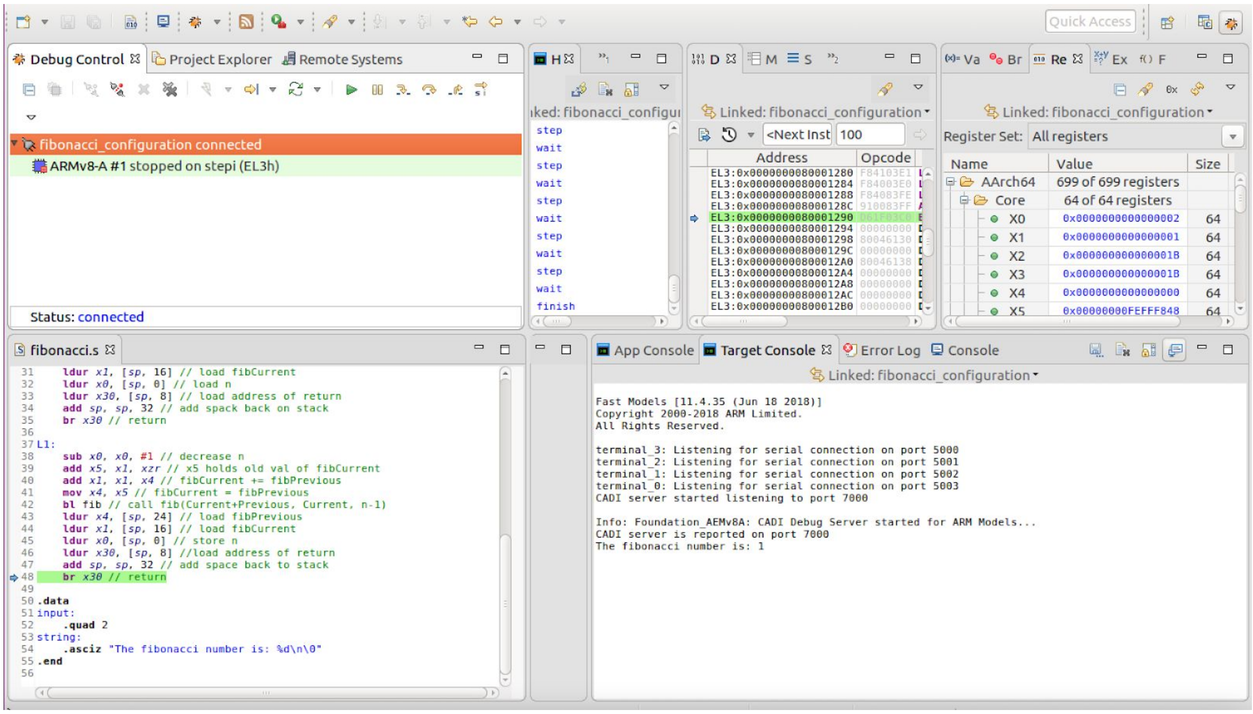
```
Name          Value                   Size
AArch64       699 of 699 registers
  Core        64 of 64 registers
    X0        0x0000000000000000      64
    X1        0x0000000000000001      64
    X2        0x0000000080046130      64
    X3        0x00000000FEFFFFF0      64
    X4        0x00000000FEFFFFE0      64
    X5        0x0000000000000000      64
```

fibonacci.s

```
 *      Author: romarazdan
 */

.text
.global main
.extern printf

main:
    .global fib
    add x1, xzr, xzr
    add x1, x1, #1 // initializes fibCurrent in reg x1 to 1
    ldr x2, =input // x2 holds address of the input
    ldr x0, [x2] // x0 holds val of n
    add x4, xzr, xzr // initializes fibPrevious in reg x4 to 0

fib:
    sub sp, sp, 32 // subtract space on stack
    stur x4, [sp, 24] // store fibPrevious
    stur x1, [sp, 16] // store fibCurrent
    stur x30, [sp, 8] // store address of return
    stur x0, [sp, 0] // store n
    subs x3, x0, #1 // check if n == 1
    b.ne L1 // if n != 1, then branch to L1, else return fibCurrent
    ldr x0, =string // loads ascii
    bl printf // prints fibCurrent
    ldur x4, [sp, 24] // load fibPrevious
```

Linked: fibonacci_configuration

```
Fast Models [11.4.35 (Jun 18 2018)]
Copyright 2000-2018 ARM Limited.
All Rights Reserved.

terminal_3: Listening for serial connection on port 5000
terminal_2: Listening for serial connection on port 5001
terminal_1: Listening for serial connection on port 5002
terminal_0: Listening for serial connection on port 5003
CADI server started listening to port 7000

Info: Foundation_AEMv8A: CADI Debug Server started for ARM Models...
CADI server is reported on port 7000
```



fibonacci_configuration connected
ARMv8-A #1 stopped on stepi (EL3h)

Status: connected

```
ıked: fibonacci_configuı
step
wait
step
wait
step
wait
step
wait
step
wait
finish
```

Linked: fibonacci_configuration

<Next Inst 100

```
Address                     Opcode
EL3:0x0000000080001280      F84103E1
EL3:0x0000000080001284      F84003E0
EL3:0x0000000080001288      F84083FE
EL3:0x000000008000128C      910083FF
EL3:0x0000000080001290      D61F03C0
EL3:0x0000000080001294      00000000
EL3:0x0000000080001298      80046130
EL3:0x000000008000129C      00000000
EL3:0x00000000800012A0      80046138
EL3:0x00000000800012A4      00000000
EL3:0x00000000800012A8      00000000
EL3:0x00000000800012AC      00000000
EL3:0x00000000800012B0      00000000
```

Linked: fibonacci_configuration

Register Set: All registers

```
Name          Value                   Size
AArch64       699 of 699 registers
  Core        64 of 64 registers
    X0        0x0000000000000002      64
    X1        0x0000000000000001      64
    X2        0x000000000000001B      64
    X3        0x000000000000001B      64
    X4        0x0000000000000000      64
    X5        0x00000000FEFFF848      64
```

fibonacci.s

```
    ldur x1, [sp, 16] // load fibCurrent
    ldur x0, [sp, 0] // load n
    ldur x30, [sp, 8] // load address of return
    add sp, sp, 32 // add spack back on stack
    br x30 // return

L1:
    sub x0, x0, #1 // decrease n
    add x5, x1, xzr // x5 holds old val of fibCurrent
    add x1, x1, x4 // fibCurrent += fibPrevious
    mov x4, x5 // fibCurrent = fibPrevious
    bl fib // call fib(Current+Previous, Current, n-1)
    ldur x4, [sp, 24] // load fibPrevious
    ldur x1, [sp, 16] // load fibCurrent
    ldur x0, [sp, 0] // store n
    ldur x30, [sp, 8] //load address of return
    add sp, sp, 32 // add space back to stack
    br x30 // return

.data
input:
    .quad 2
string:
    .asciz "The fibonacci number is: %d\n\0"
.end

```

Linked: fibonacci_configuration

```
Fast Models [11.4.35 (Jun 18 2018)]
Copyright 2000-2018 ARM Limited.
All Rights Reserved.

terminal_3: Listening for serial connection on port 5000
terminal_2: Listening for serial connection on port 5001
terminal_1: Listening for serial connection on port 5002
terminal_0: Listening for serial connection on port 5003
CADI server started listening to port 7000

Info: Foundation_AEMv8A: CADI Debug Server started for ARM Models...
CADI server is reported on port 7000
The fibonacci number is: 1
```