Experiment 2:
MCSDK demos

August 18, 2011

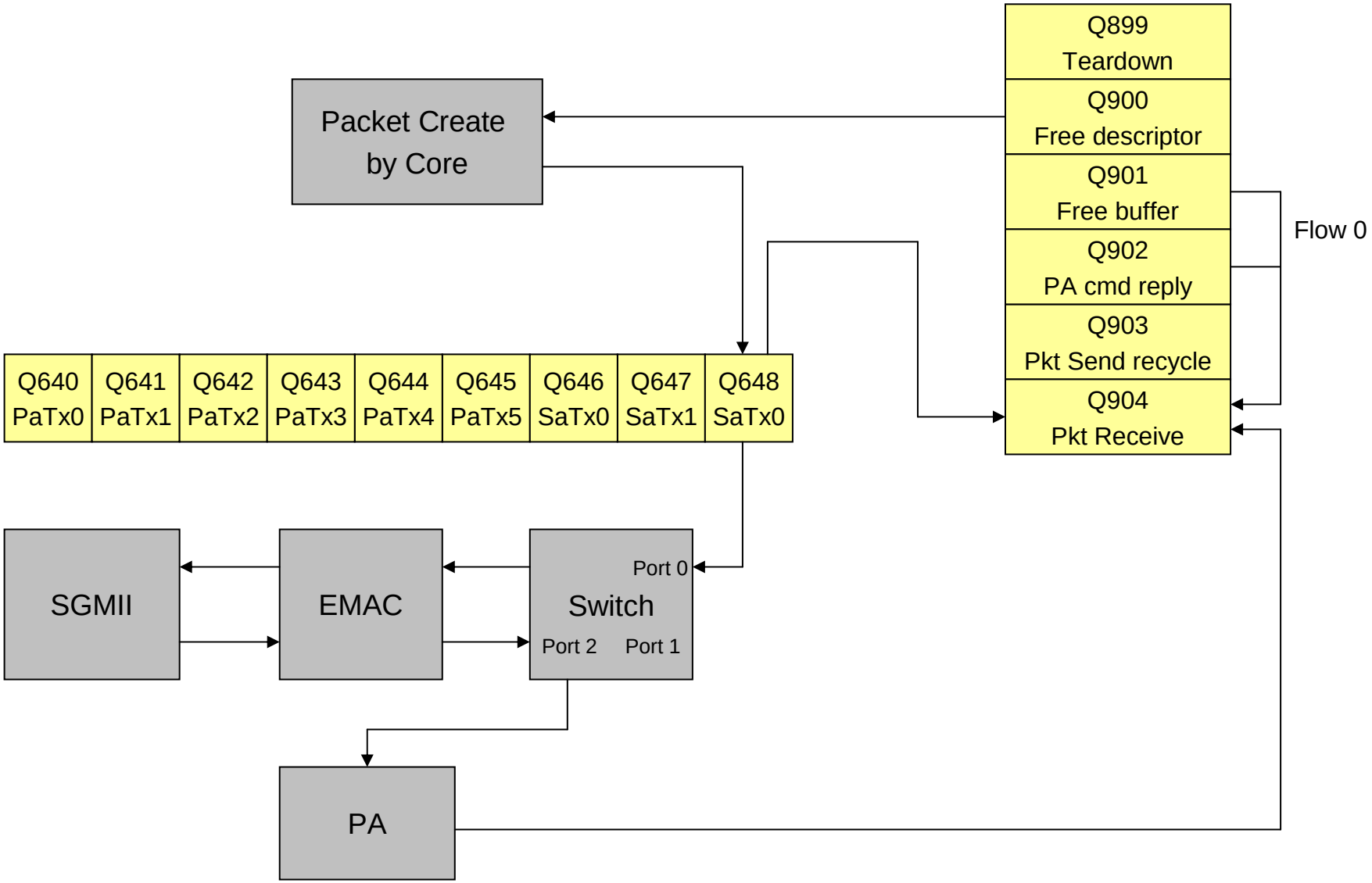
Texas Instruments

Emulation environment

- Shannon EVM
- XDS100v1 USB Emulator

Demo 1:
PA Simple Example

Flow Chart

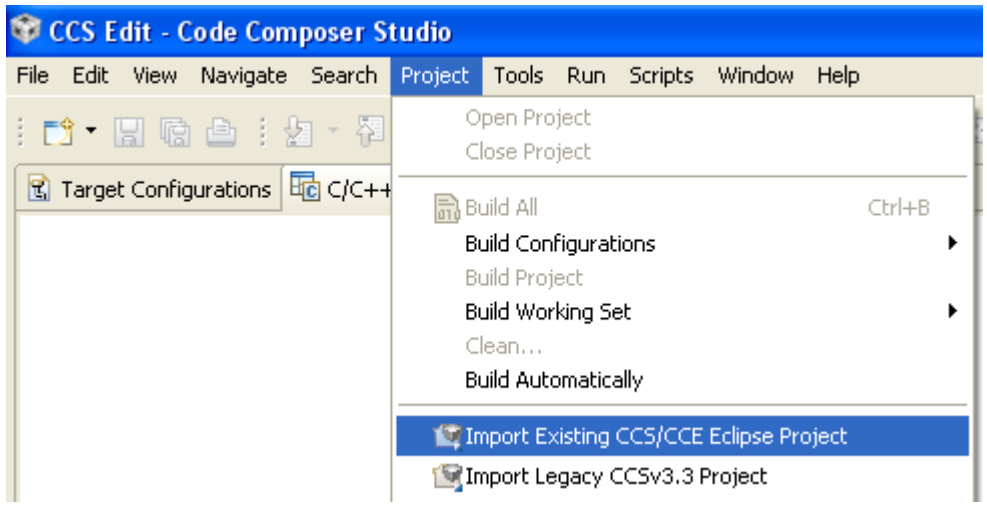

Q899
Teardown
Q900
Free descriptor
Q901
Free buffer
Q902
PA cmd reply
Q903
Pkt Send recycle
Q904
Pkt Receive
Flow 0
Packet Create
by Core
Q640
PaTx0
Q641
PaTx1
Q642
PaTx2
Q643
PaTx3
Q644
PaTx4
Q645
PaTx5
Q646
SaTx0
Q647
SaTx1
Q648
SaTx0
SGMII
EMAC
Port 0
Switch
Port 2
Port 1
PA

PA Simple Project:

- Import CCS project from PDK example
- Build project
- Specify emulation configuration
  - Target select
  - Add evmc6678l.gel file
- Launch target configuration
- Load program and run

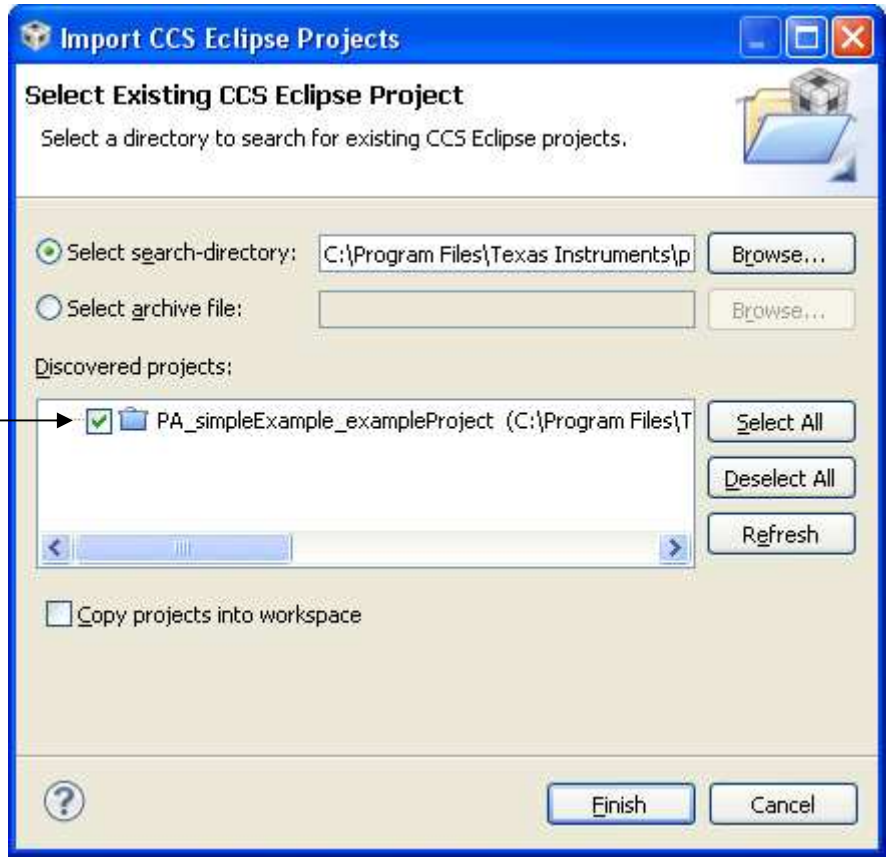
Import PDK example

- Import project
  - Project ->Import existing CCS eclipse project



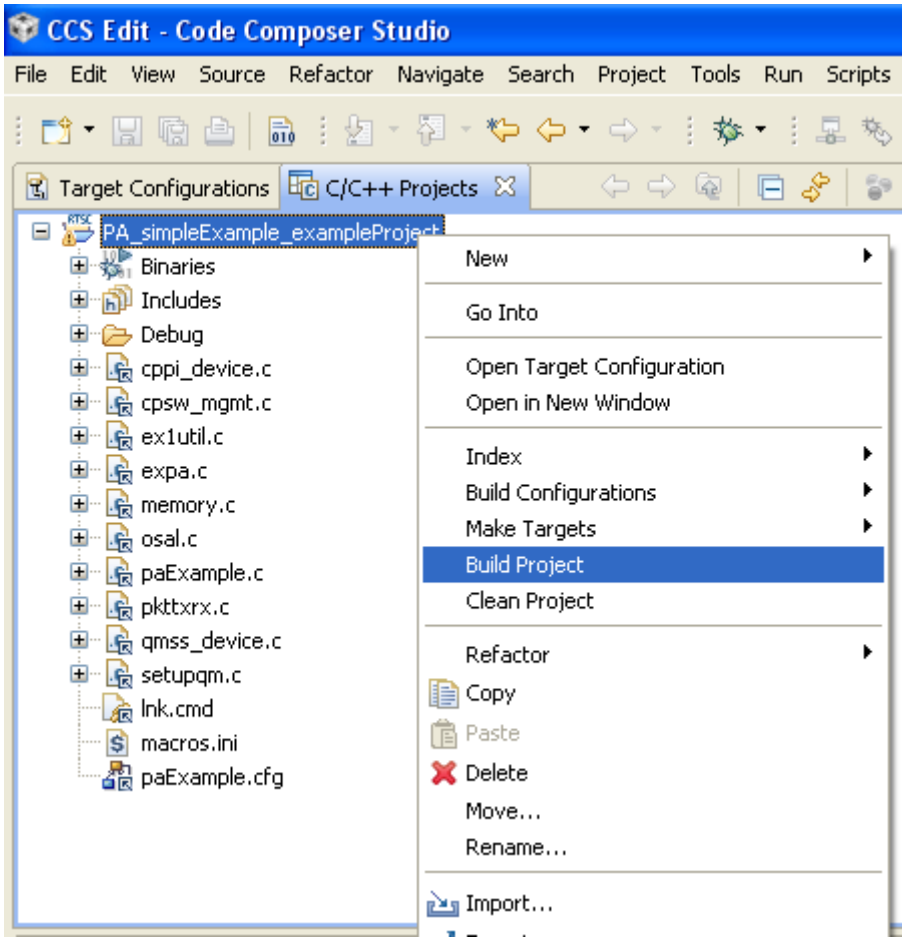
- Browse to the directory

  (C:\Program Files\Texas Instruments\pdk_C6678_1_0_0_11\packages\ti\drv\exampleProjects\PA_simpleExample_exampleProject)
- PA Simple Example
- Select ‘Finish’ when done



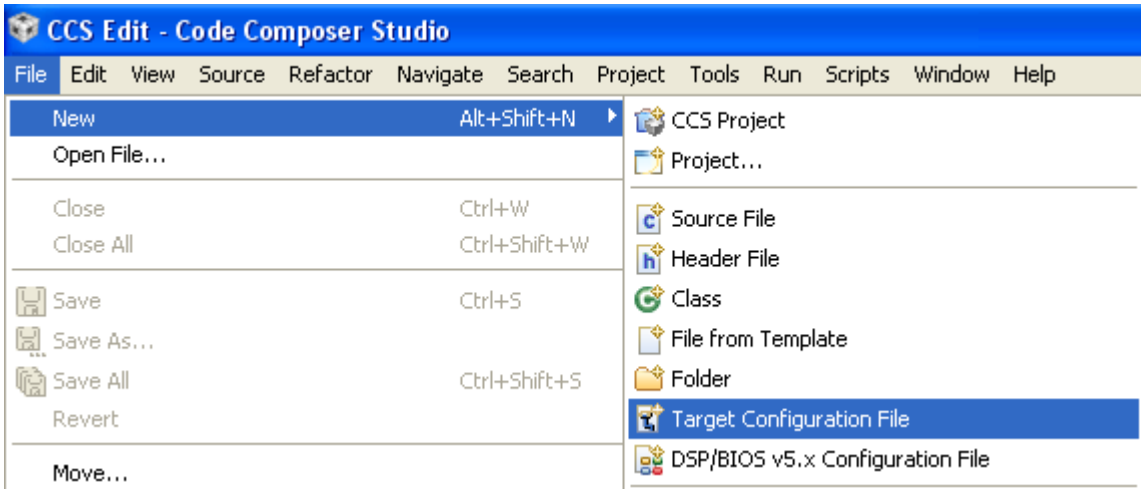
Build project

- Right-click the project and select ‘build project’

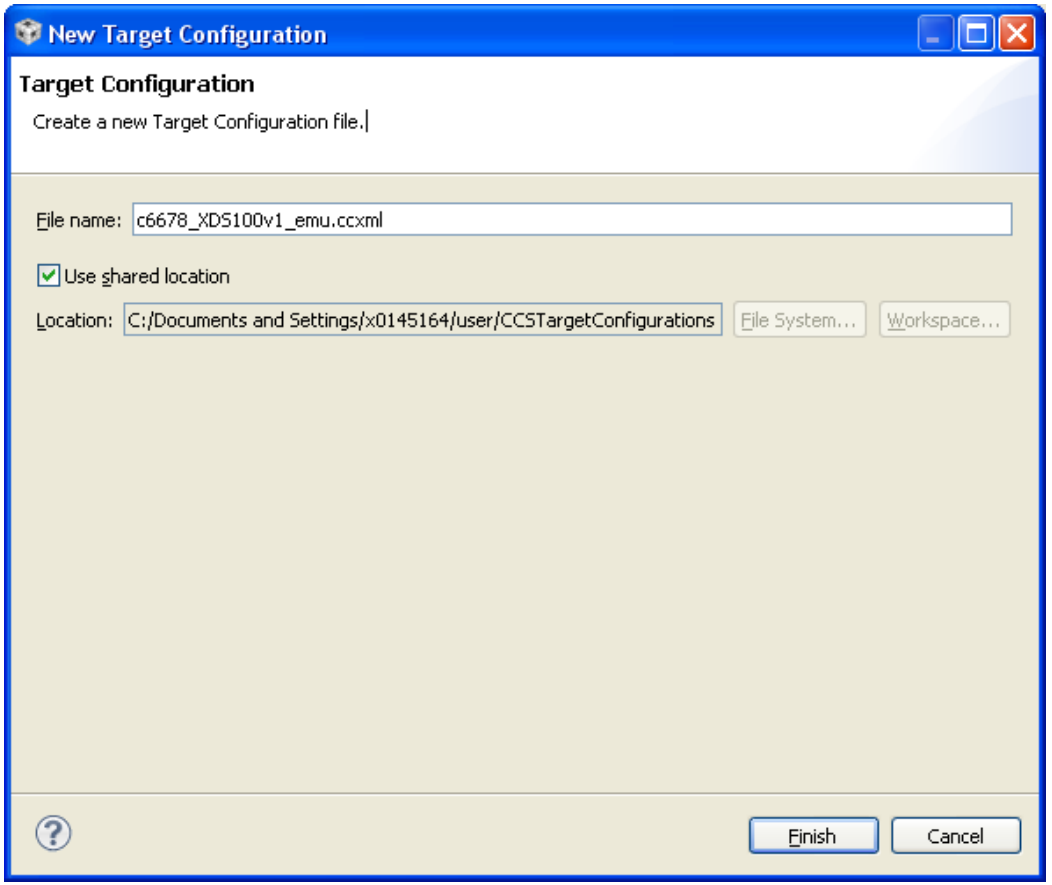


Specify emulation configuration

- Add a new target configuration
  - File->New->Target configuration File



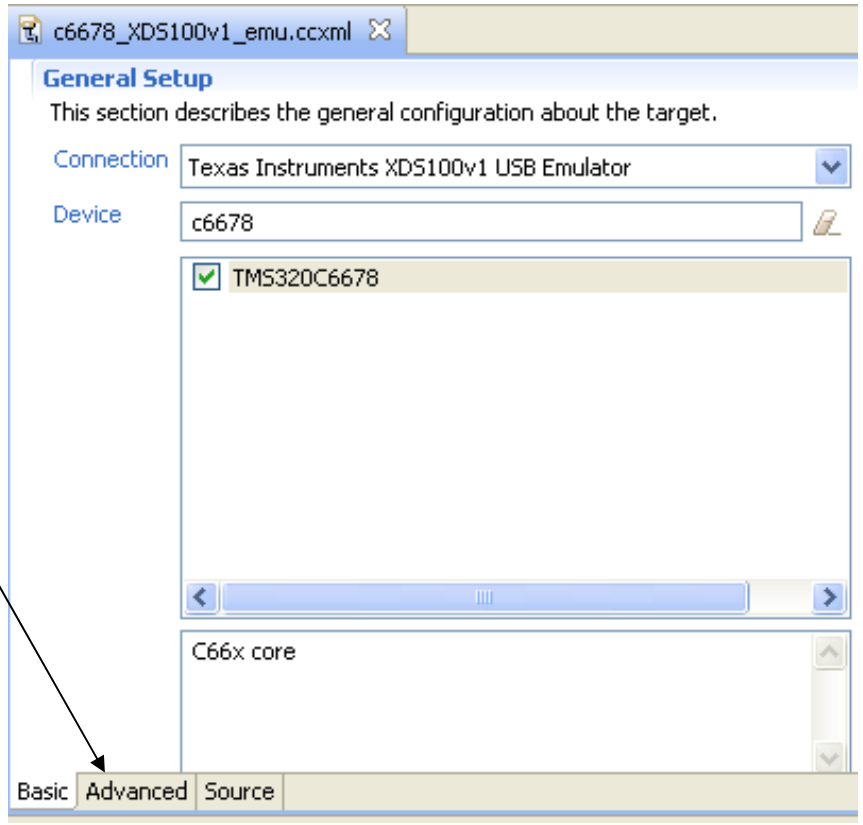
- Provide a name for the new target



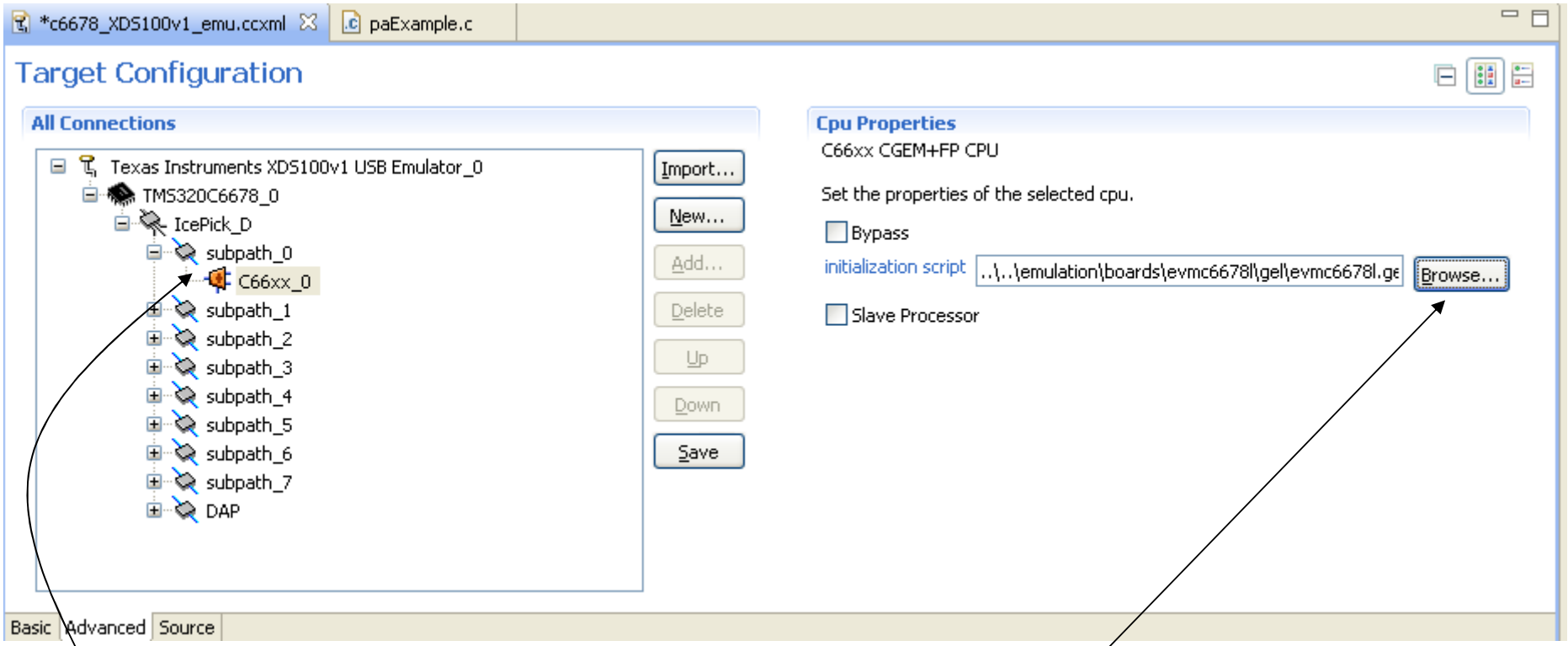
- Select ‘Use shared location’
- Select ‘Finish’ when done

Target select

- Use drop down menu to select the connection type
- Select the device
- Save the file when done
- Select ‘Advanced’ tab

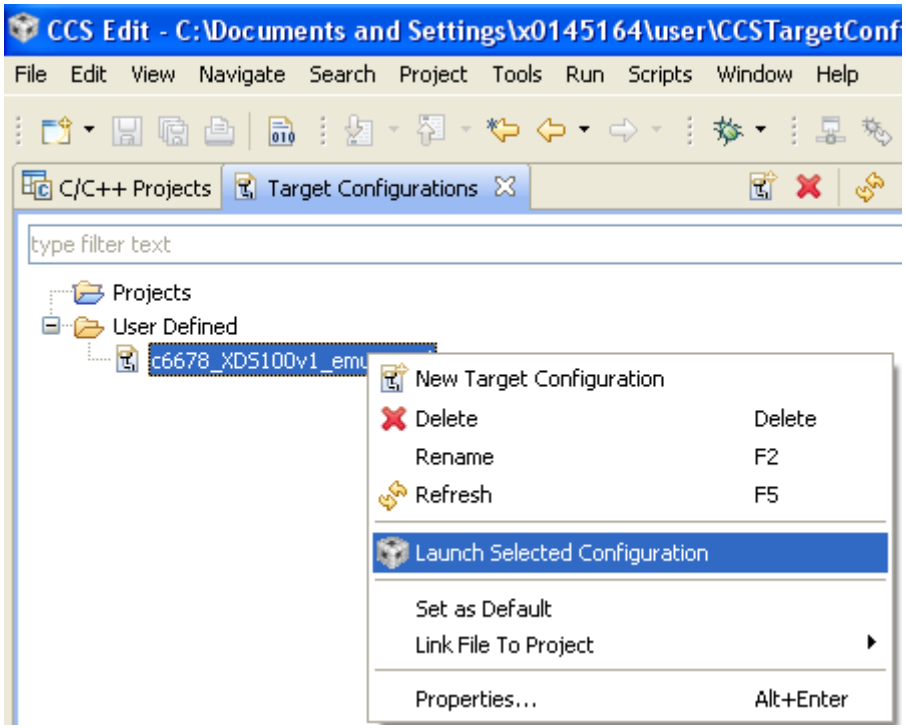


Add evmc6678l.gel file



- Select core 0
- Browse to select evm gel file
  (C:\Program Files\Texas Instruments\ccsv5\ccs_base_5.0.3.00028\
  emulation\boards\evmc6678l\gel\evmc6678l.gel)
- Save the file when done

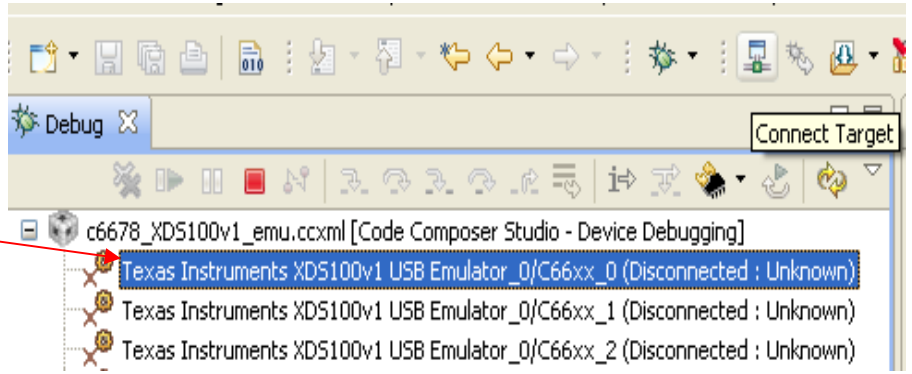
Launch target configuration



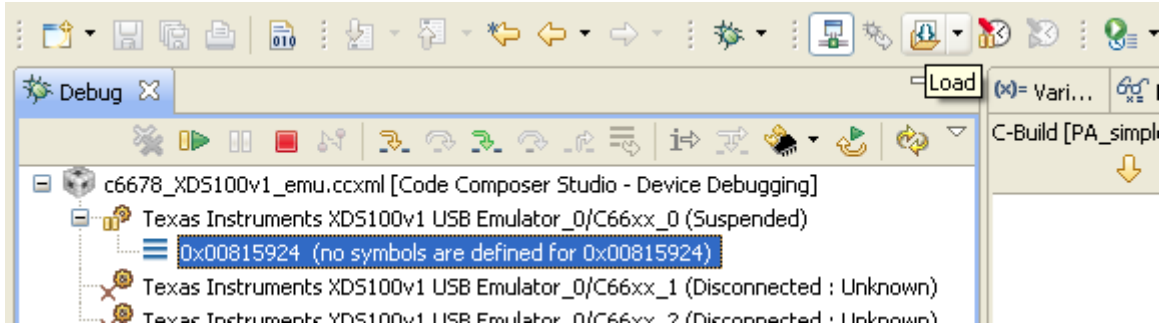
- Switch to target configuration window
- Right-click on c6678_XDS100v1_emu.ccxml, select ‘launch selected configuration’

Load program and run

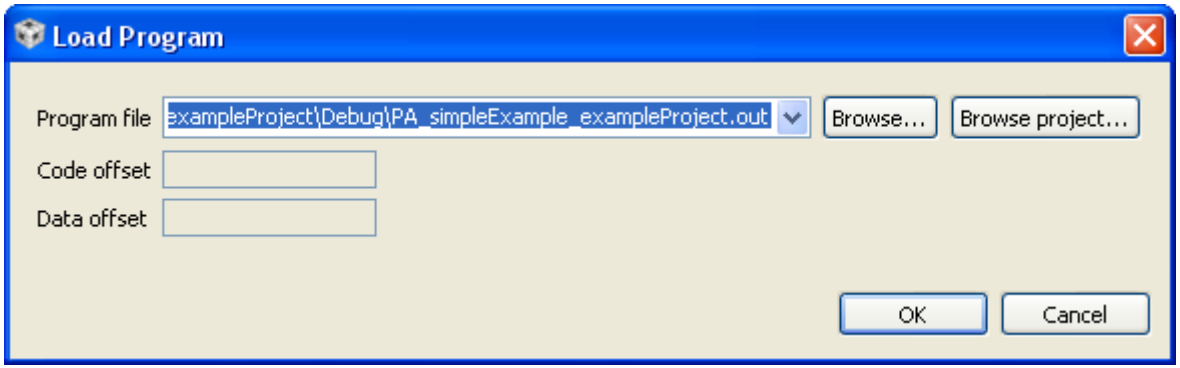
- Select core 0
- Click ‘connect target’



- Load program

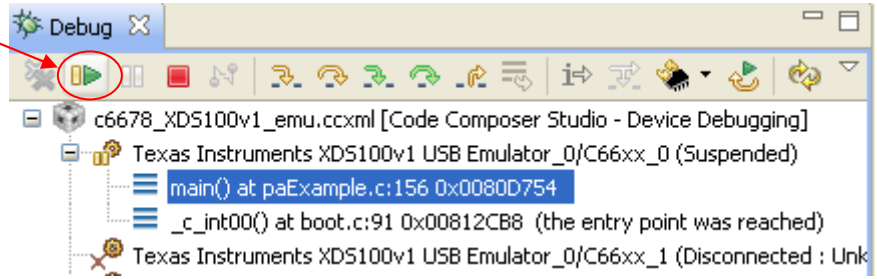


- Browse for *.out file



- When program is loaded,

Select the ‘run’ button to execute application



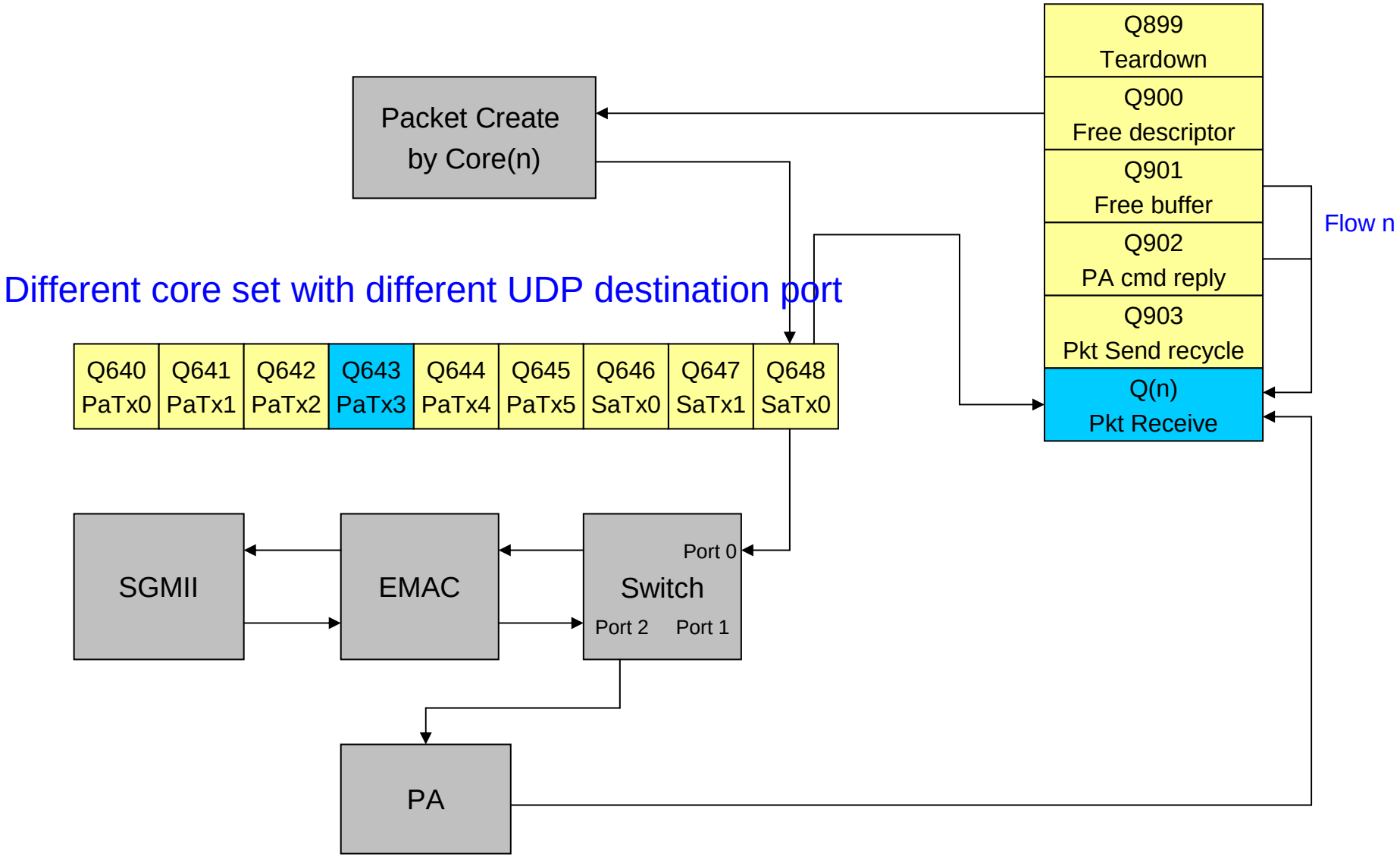
Core0 CIO output

c6678_XDS100v1_emu.ccxml:C66xx_0:CIO

```
function findPacket: Correct packet found
--- PA STATS ---
C1 number of packets:           4
C1 number IPv4 packets:         1
C1 number IPv6 packets:         0
C1 number custom packets:       0
C1 number non IP packets:       0
C1 number llc/snap fail:        0
C1 number table matched:        2
C1 number failed table matched: 0
C1 number IP frags:             0
C1 number IP depth overflow:    0
C1 number vlan depth overflow:  0
C1 number gre depth overflow:   0
C1 number mpls packets:         0
C1 number of parse fail:        0
C1 number invalid IPv6 opts:    0
C1 number of command failures:  0
C1 number invalid reply dests:  0
C1 number of silent discard:    0
C1 number of invalid control:   0
C1 number of invalid states:    0
C1 number of system fails:      0

C2 number of parse fail:          0
C2 number invalid headers:        0
C2 number of UDP packets:         1
C2 number of TCP packets:         0
C2 number of custom packets:      0
C2 number of command failures:    0
C2 number of invalid reply dests: 0
C2 number of silent discard:      0
C2 number of invalid control:     0

Modify number of command file: 0

Common number of ID allocation fail: 0

Test Completed successfully
```

Demo 2:
PA Multi-core Example

Flow Path


Q899
Teardown
Q900
Free descriptor
Q901
Free buffer
Q902
PA cmd reply
Q903
Pkt Send recycle
Q(n)
Pkt Receive
Flow n
Packet Create
by Core(n)
Different core set with different UDP destination port
Q640
PaTx0
Q641
PaTx1
Q642
PaTx2
Q643
PaTx3
Q644
PaTx4
Q645
PaTx5
Q646
SaTx0
Q647
SaTx1
Q648
SaTx0
SGMII
EMAC
Port 0
Switch
Port 2
Port 1
PA

Flow Path

- **Initialize the components required to run the example:**
  - Queue Manager (QM) Subsystem
  - Packet Accelerator (PA) CPPI DMA
  - Ethernet Subsystem (Ethernet switch + SGMII + MDIO)
  - PA Subsystem + PDSP

- **Sets up the CPPI descriptors and Queues required for sending and receiving data using Ethernet.**
  - Uses Host descriptors
  - Uses High Priority Accumulation interrupts

- **Sets up the example application's configuration (MAC address it uses to send/receive data; IP address and port number it's listening on) in PA Subsystem so as to enable the PASS to forward all packets matching this configuration onto the application for processing.**
  - Switch MAC address configured = 0x10:0x11:0x12:0x13:0x14:0x15
  - Example's IP address = 192.168.1.10
  - Example App's listening port = 0x5678 + corenum

Flow Path (continued)

- **Sends packets onto wire (constructed manually in code here with following settings):**
  - Source MAC = 0x00:0x01:0x02:0x03:0x04:0x05
    Destination MAC = 0x10:0x11:0x12:0x13:0x14:0x15
  - Source IP = 192.168.1.1
    Destination IP = 192.168.1.10
  - Source Port = 0x1234
    Destination Port= 0x5678 + corenum
  - Payload Data (80 bytes)

  The packets sent by the application are sent onto wire and since the destination MAC on the packet is the Ethernet Switch MAC address, the packets are received by simulator and passed back up to the example application for processing.

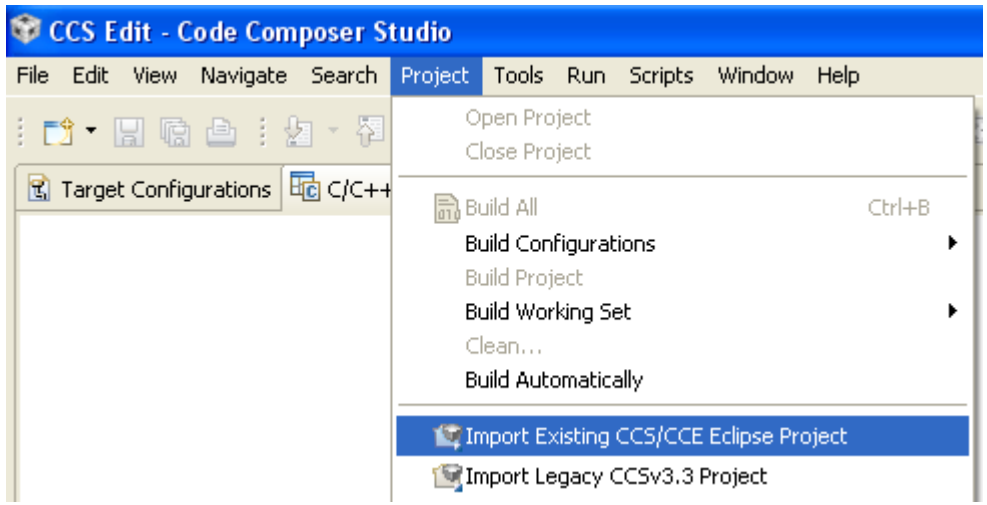
- **Application receives all packets using QM High priority interrupt registered; Validates received packet against data sent.**

PA Multi-core Project:

- Import CCS project from PDK example
- Build project
- Launch target configuration
- Load program and run

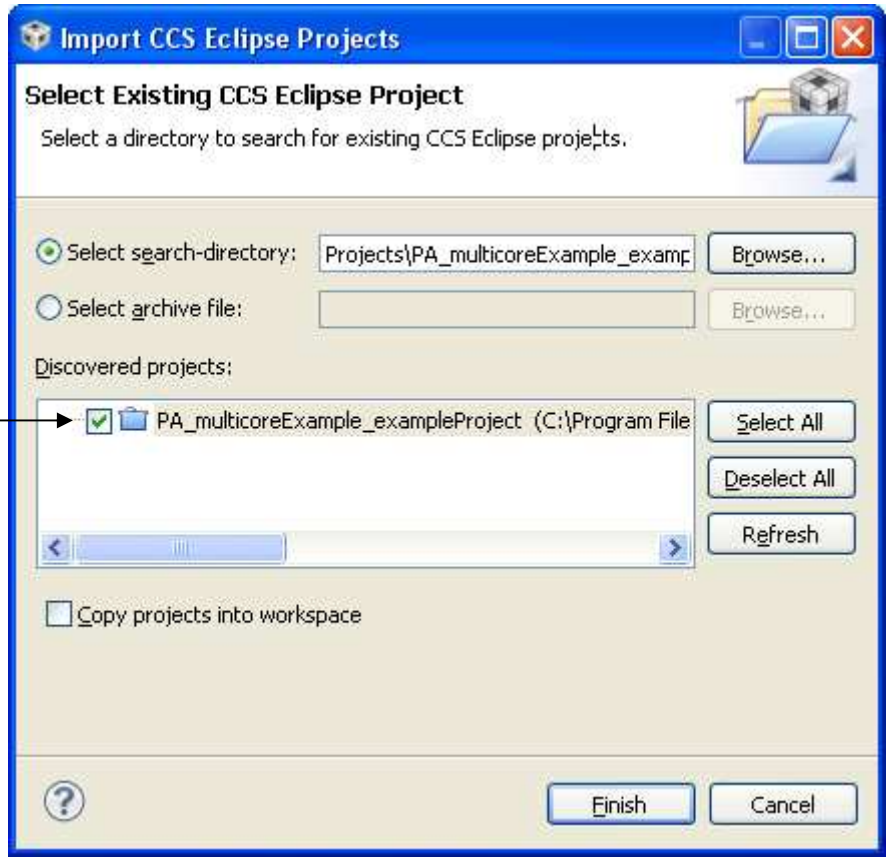
Import PDK Example

- Import project
  - Project ->Import existing CCS eclipse project



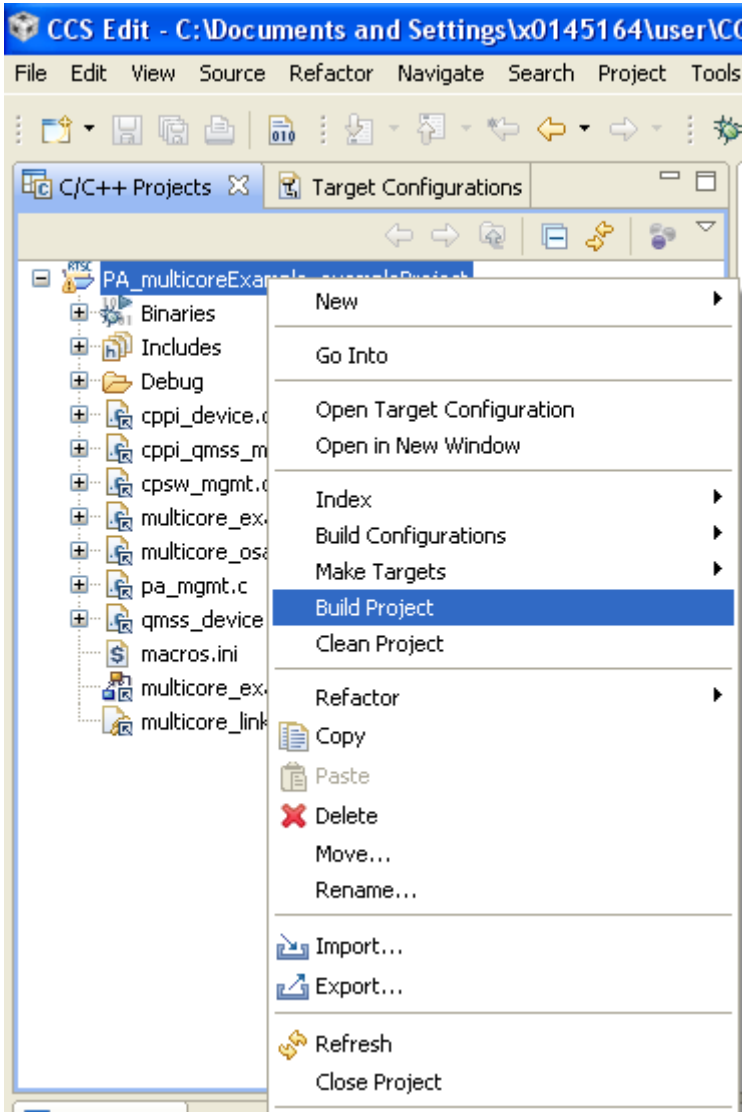
- Browse to the directory

  (C:\Program Files\Texas Instruments\pdk_C6678_1_0_0_11\packages\ti\drv\exampleProjects\PA_multicoreExample_exampleProject)
- PA_multicore example
- Select ‘Finish’ when done

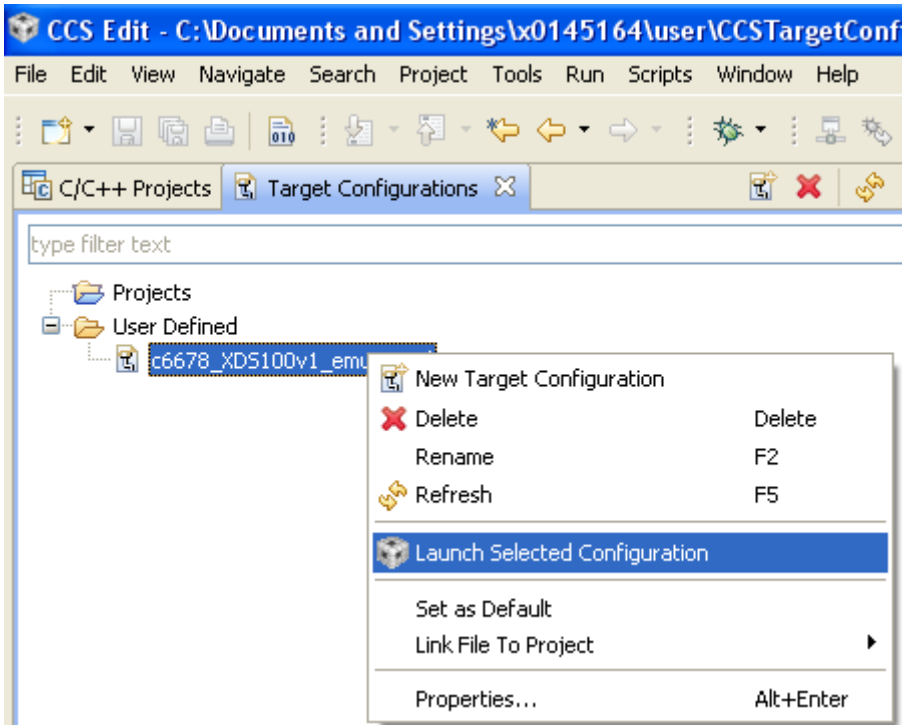


Build project

- right-click the project and select ‘build project’



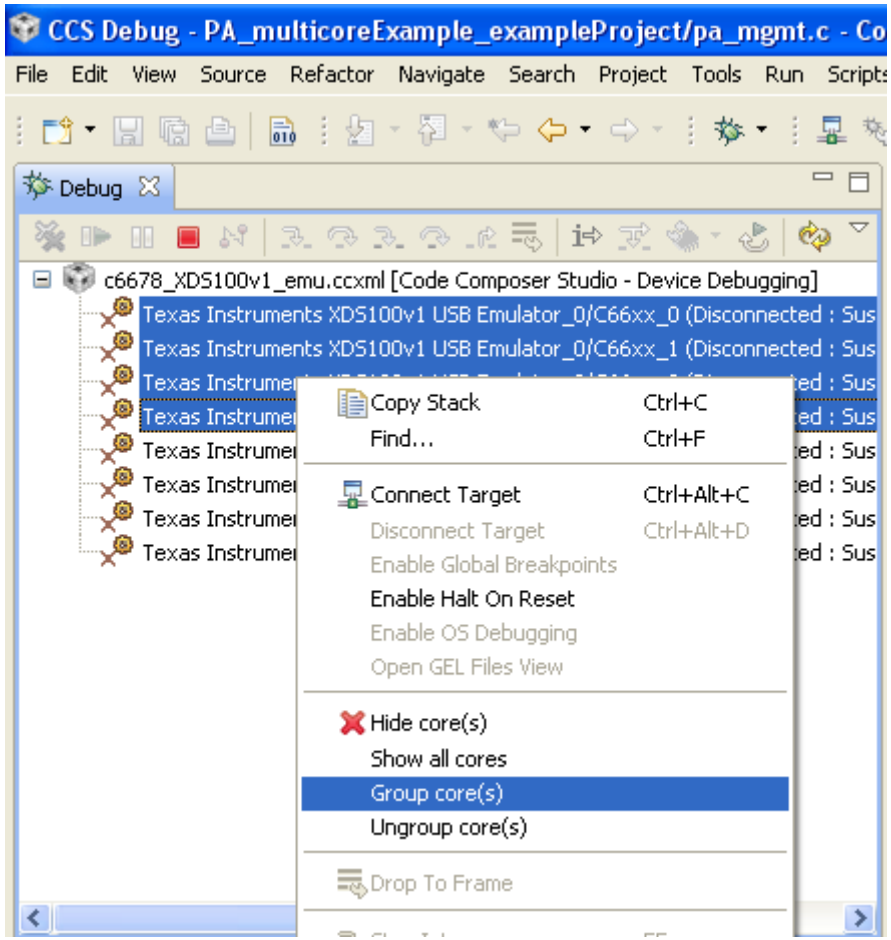
Launch target configuration



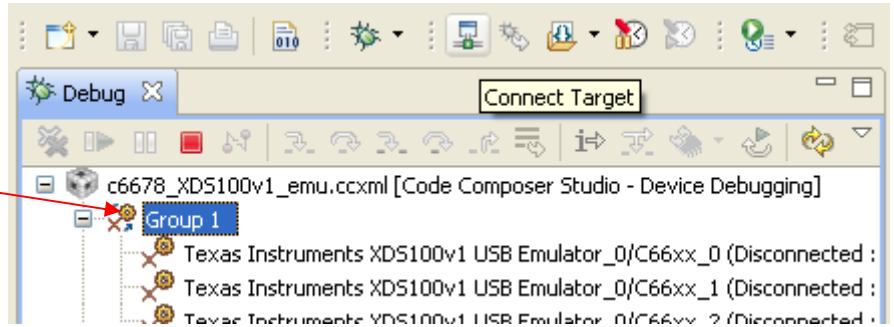
- Switch to target configuration window
- Right-click on c6678_XDS100v1_emu.ccxml, select ‘launch selected configuration’

Load program and run

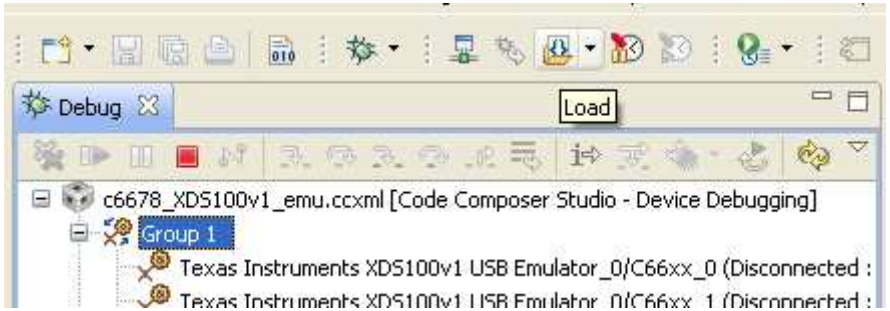
- Group core 0~3



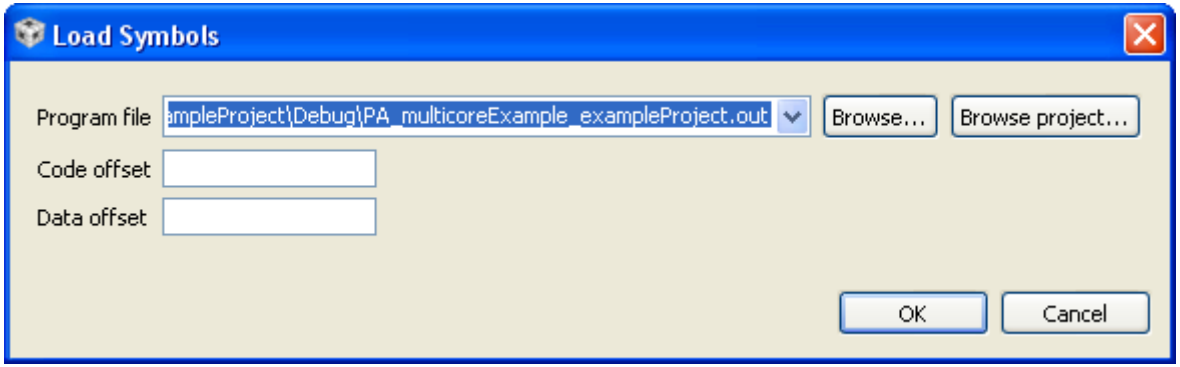
- Select group 1
- Select ‘connect target’



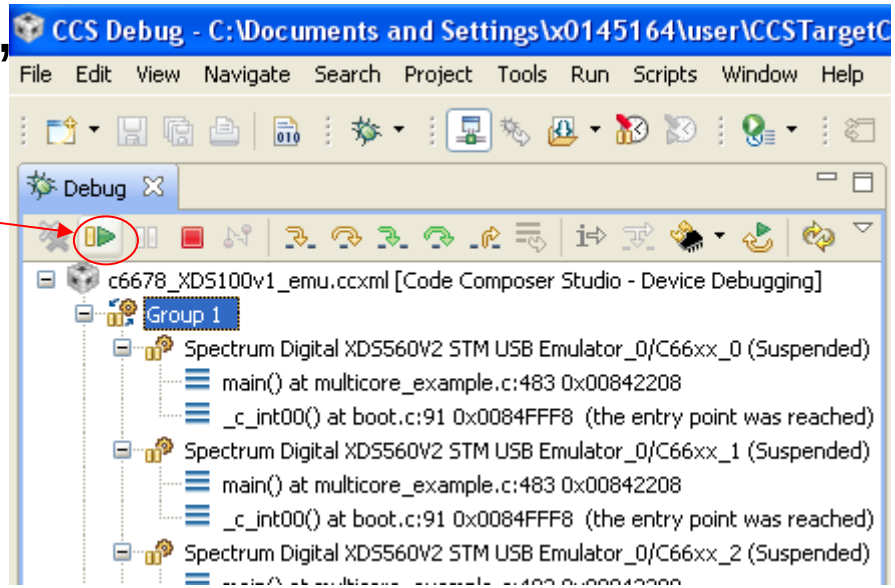
- Load program



- Browse for *.out file

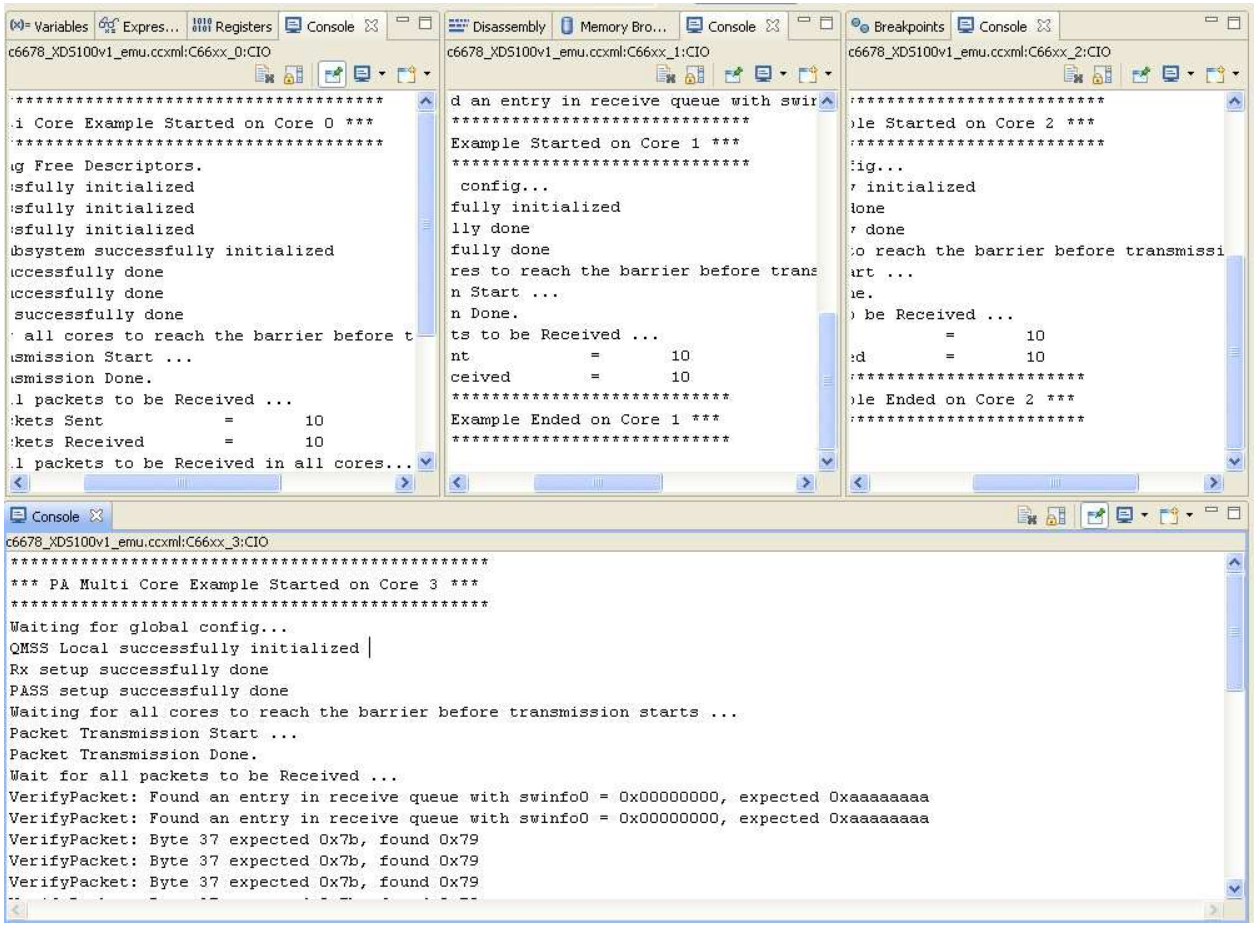


- When program is loaded, Select the ‘run’ button to execute application on Group 1



Result

- Open 4 consoles to show 4 cores result together



Thanks!

August 17, 2011