Verification of Ethernet Passthrough to Android in TML Gen3 target

1. Enable Ethernet Pass-through to Android: -

QHX Gerrits: -

Topic: https://gerrit1.harman.com/q/topic:TML_Gen3_Ethernet_Passthrough

Sl. No.	MR link	Description	Remarks
1.	https://gerrit1.harman.com/c/GTC_Foundation_Software/Integration/QNX/Platform/HQX/BSP/+/279673/	Ethernet pass through to IVI	

Android Gerrits: -

Topic: https://androidhub.harman.com/q/topic:TML_Gen3_Ethernet_Passthrough

Sl. No.	MR link	Description	Remarks
1.	https://androidhub.harman.com/c/kernel/msm-5.4/+/444921	Merged STMMAC changes from TML branch	
2.	https://androidhub.harman.com/c/kernel/msm-5.4/+/444934	Integrate Marvell 88Q111X PHY driver.	
3.	https://androidhub.harman.com/c/kernel/msm-5.4/+/444937	Add EMAC clocks in virtio clock driver	
5.	https://androidhub.harman.com/c/kernel/msm-5.4/+/446836	Add dts property to EMAC driver for ethernet interface name	
6.	https://androidhub.harman.com/c/kernel/msm-5.4/+/456961	Configure ethernet PHY - 88Q111X as master to communicate with TCU	
7.	https://androidhub.harman.com/c/kernel/msm-5.4/+/462073	Add mutex lock to make sysfs MDIO access atomic	To Test

Android Device Tree Gerrit

Sl. No.	MR Link	Description	Remark
1.	https://androidhub.harman.com/c/vendor/qcom/proprietary/+/444788	Add dts changes for Ethernet Controller driver	
2.	https://androidhub.harman.com/c/vendor/qcom/proprietary/+/446820	Add dts change to change Ethernet interface name	
3.	https://androidhub.harman.com/c/vendor/qcom/proprietary/+/456952	DTS change to set ethernet PHY-88Q111x as master	

2. Addition of conditional build flag: -

QHX Gerrits: -

https://gerrit1.harman.com/q/topic:TML_ETH_BUILD_FLAG

Sl. No.	MR link	Description	Remarks
1.	https://gerrit1.harman.com/c/GTC_Foundation_Software/Integration/QNX/Platform/HQX/BSP/+/283840	Makefile changes to modify variant config file	

2.	https://gerrit1.harman.com/c/GTC_Foundation_Software/Integration/QNX/Platform/HQX/BSP/+/283867	Add compilation flag to allow EMAC clock to IVI	
3.	https://gerrit1.harman.com/c/GTC_Foundation_Software/Integration/QNX/Platform/HQX/BSP/+/283875	Add SMMU related XML file to passthough Ethernet on IVI	
4.	https://gerrit1.harman.com/c/Digital_Cluster_Cockpit/Integration/OEM/TATAGen3/athena_gl/+/283979	Add flag to enable Ethernet on Android	

Android Kernel Gerrit: -

https://androidhub.harman.com/q/topic:TML_ETH_BUILD_FLAG_IVI

Sl. No.	MR link	Description	Remarks
1.	**https://androidhub.harman.com/c/kernel/msm-5.4/+/459342**	**Add config fragment file to enable Ethernet device**	

Android Device Tree Gerrits: -

https://androidhub.harman.com/q/topic:TML_ETH_BUILD_FLAG_IVI

Sl. No.	MR link	Description	Remarks
1.	**https://androidhub.harman.com/c/vendor/qcom/proprietary/+/449590**	**Add conditional flag in DTS Makefile to enable Ethernet**	
2.	**https://androidhub.harman.com/c/vendor/qcom/proprietary/+/459391**	**Add conditional flag in DTS Makefile**	**Both MRs in Sl. No. 1 and 2 are to be sqaushed.**

Platform Build Gerrit: -

https://androidhub.harman.com/q/topic:TML_ETH_BUILD_FLAG_IVI

Sl. No.	MR link	Description	Remarks
1.	**https://androidhub.harman.com/c/platform/build/+/459335**	**Add ETHERNET_IVI flag**	

3. Ethernet Tests

Switching between Android and QNX console: -

TML_Gen3_Procedure.txt

3.1 Ping test

Command to run on target: -

```
## Enable ethernet interface

ifconfig ethPhy0 192.168.2.5 up

## Flush IP table
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT

## Then flush the rules:
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

## Ping test PC
ping -I ethPhy0 192.168.2.6
```

3.2 Iperf throughput test

Command to run on target: -

```
## iperf testing on TML target

export TMPDIR=/data/local/tmp
export TMPDIR=/data/local/tmp
export TEMP=/data/local/tmp
cd /data/local/tmp
iperf3 -s
```

Command to run on test PC: -

```
 # Make ethernet interface Private network.

 > Get-NetConnectionProfile
 > Set-NetConnectionProfile -InterfaceIndex <interface_index> -NetworkCategory Private

 # iperf command: -
 iperf3.exe -c 192.168.2.5 -R -t 60
 Output: -
Connecting to host 192.168.2.5, port 5201
Reverse mode, remote host 192.168.2.5 is sending
[  4] local 192.168.2.6 port 50045 connected to 192.168.2.5 port 5201
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.00   sec  11.4 MBytes  96.0 Mbits/sec
[  4]   1.00-2.00   sec  11.2 MBytes  93.9 Mbits/sec
[  4]   2.00-3.00   sec  11.0 MBytes  92.1 Mbits/sec
[  4]   3.00-4.00   sec  11.1 MBytes  93.4 Mbits/sec
[  4]   4.00-5.00   sec  11.2 MBytes  93.8 Mbits/sec
//...
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-60.00  sec   666 MBytes  93.1 Mbits/sec  100             sender
[  4]   0.00-60.00  sec   665 MBytes  93.0 Mbits/sec                  receiver
```

4. CPU load analysis due to Ethernet using DMIPS calculation

4.1. CPU core assignment to QNX and Android

File: - apps/qnx_ap/target/hypervisor/host/build_files/system.build.tmpl

CPU core assigned to QNX: -

```
#if defined(__GUEST_TYPE_LV__)

#ifdef VARIANT_6155
cpu sched 10 runmask 6,7
cpu sched 10 runmask 6,7
cpu sched 10 runmask 0,1,2,3,4,5
cpu sched 10 runmask 0,1,2,3,4,5
cpu sched 10 runmask 0,1,2,3,4,5
cpu sched 10 runmask 0,1,2,3,4,5
cpu sched 10 runmask 0,1,2,3,4,5
cpu sched 10 runmask 0,1,2,3,4,5
#endif
//....


#if defined(__GUEST_TYPE_LA__)
//....
#ifdef VARIANT_6155
cpu sched 25 runmask 6,7
cpu sched 25 runmask 7,6
cpu sched 25 runmask 5,4,3
cpu sched 25 runmask 4,3,5
cpu sched 25 runmask 3,5,4
#endif
//....
```

__GUEST_TYPE_LA___ => Android

__GUEST_TYPE_LV___ => QNX

From above, CPU cores assigned to QNX are 0, 1, 2, 3, 4, 5, 6, 7.

CPU cores assigned to Android are 3, 4, 5, 6, 7.

4.2. DMIPS values of QC CPU cores

Qualcomm 6150 has 2xGOLD & 6xSILVER KYRO cores.

For QC6150 SOC, Gold cores work at 1.7GHz and Silver cores work at 1.0GHz.

	DMIPS/Mhz	QC6145 (1.0/0.8Ghz)	QC6150 (1.7/1.0Ghz)	QC6155 (1.9/1.6Ghz)
Gold	6.57	13,140	22,338	24,966
Silver	2.66	12,768	15,960	25,536
Total		25,908	38,298	50,502

DMIPS/MHz for Gold core = 6.57

Maximum DMIPS per Gold core @1.7GHz = (CPU Clock Frequency in MHz*)* x (CPU DMIPS value in per MHz) = (1.7x1000) x 6.57 = 11,169

So, total DMIPS offered by 2 Gold cores @1.7GHz= 11,169 x 2 = 22,238.

DMIPS/MHz for Silver core = 2.66

Maximum DMIPS per Silver core @1.0GHz = *(Max. Freq. of operation in MHz)* x DMIPS/MHz = (1.0x1000) x 2.66 = 2,660

So, total DMIPS offered by 6 Silver cores @1.7GHz= 11,169 x 2 = 15,960.

4.3. Method I. : DMIPS measured by Network Iperf throughput test

I. Ensure Ethernet Passthrough to Android is enabled. If possible verify target is pinging to test PC, using test steps mentioned in Section - 3.1.

II. On Android console run Iperf command to communicate with test PC, using test steps mentioned in section - 3.2.

III. Switch to QNX console and run top command, as below.

```
# top -d -b -i30 -z5 -D1 -t
O/p -

Computing times...94 processes; 963 threads;
CPU states: 20.3% user, 0.3% kernel
CPU  0 Idle: 79.5%
CPU  1 Idle: 89.1%
CPU  2 Idle: 91.4%
CPU  3 Idle: 79.2%
CPU  4 Idle: 75.1%
CPU  5 Idle: 80.6%
CPU  6 Idle: 68.1%
CPU  7 Idle: 71.1%
Memory: 6138M total, 235M avail, page size 4K
```

Note -

Average CPU load = (100 - CPU Idle %)

IV. Without running "iperf" command between test PC and target, run same top command on QNX console, as shown in step III.

V. Based on CPU load, table as below is prepared, with and without iperf running on target.

Android Physical CPU Core#	Average CPU load without running iperf on Android	Average CPU load with iperf running on Android	Increase in CPU load
3	21%	22%	1%
4	20%	22%	2%
5	21%	21%	0%
6	24%	33%	9%
7	22%	32%	10%
			Total CPU load by Iperf = 22%

This readings may not be accurate, because on Physical CPU cores 3 to 7, CPU load may increase or decrease due to other services or applications. Also, Ethernet Interrupts are on different cores (vCPU-0) that is Physical CPU core - 6 and 7.

4.4. Method II. : DMIPS measured by Network Iperf throughput test by setting CPU affinity

To measure DMIPS consumed by "iperf" application, we will set affinity to "iperf" on some vCPU, for example on vCPU-03 and also we will try to set affinity to Ethernet interrupts on vCPU-03.

Steps followed has been mentioned here.

Step I. Run "iperf" command on target at Android console on target.

```
export TMPDIR=/data/local/tmp
export TMPDIR=/data/local/tmp
export TEMP=/data/local/tmp
cd /data/local/tmp

# iperf3  -s  --affinity 3 > /dev/null &
```

=> Verify CPU affinity set to vCPU-03 by running below commands.

```
# ps | grep iperf3

O/p - <PID of iperf3> ....

# taskset -p <PID of iperf3>

O/p - pid <PID>'s current affinity mask: 8
```

Note - 8 in binary 1000b, here bit-3 set to 1 indicates CPU affinity set to vCPU-03.

Step II. Set affinity of Ethernet IRQs to a CPU core, say vCPU-3.

Find IRQ number of Ethernet interrupt and set affinity of Ethernet interrupt on vCPU-03, by running below commands on Android console.

```
# cat /proc/interrupts | grep ethPhy0
O/p - 135 ....

# echo 08 > /proc/irq/135/smp_affinity
```

Step III. Run below iperf command on test PC.

```
> iperf3.exe -c 192.168.2.5 -R -t 60
```

Step IV. Run top command on QNX console, while iperf command runs for Android VM.

```
# top -d -b -i30 -z5 -D1 -t

O/p -

. . . .

                Min      Max      Average
CPU 0 idle:   77%      79%      78%
CPU 1 idle:   87%      90%      88%
CPU 2 idle:   91%      93%      92%
CPU 3 idle:   62%      68%      65%
CPU 4 idle:   60%      70%      66%
CPU 5 idle:   64%      70%      68%
CPU 6 idle:   69%      79%      75%
CPU 7 idle:   55%      79%      74%
Mem Avail:   236MB     236MB     236MB
Processes:    94       94       94
Threads:     958      958      958

. . . .
```

Step V. Run "top" command on QNX, without running "iperf" on Android and test PC.

```
# top -d -b -i30 -z5 -D1 -t

O/p -

. . . .

                Min      Max      Average
CPU 0 idle:   77%      82%      79%
CPU 1 idle:   86%      92%      89%
CPU 2 idle:   90%      94%      92%
CPU 3 idle:   62%      80%      70%
CPU 4 idle:   60%      81%      71%
CPU 5 idle:   64%      82%      73%
CPU 6 idle:   69%      80%      76%
CPU 7 idle:   55%      80%      75%
Mem Avail:   236MB     236MB     236MB
Processes:    94       94       94
Threads:     958      958      958

. . . .
```

Step VI. Calculate CPU load by "iperf" utility and interrupt coming due to Ethernet.

Physical CPU core number	CPU load without "iperf"	CPU load with "iperf" and Ethernet interrupts on vCPU-03	Change in CPU load
0	20%	22%	2%
1	11%	12%	1%
2	8%	8%	0%
3	23%	35%	12%
4	23%	34%	11%
5	21%	32%	11%
6	24%	25%	1%
7	23%	26%	3%

As described in section 4.1, vCPU-03 on Android is served by scheduling three Physical CPU cores - 3, 4 and 5. From table above, we can see that CPU load has mainly increased on Physical CPU cores 3, 4 and 5, which is nothing but vCPU-03, to which CPU affinity for "iperf" and Ethernet interrupts are set.

Total increase in CPU load by "iperf" on vCPU-03 = (12 + 11 + 11) = **34%**

Step VII. DMIPS consumed by "iperf".

For QC SOC: SA6150P, CPU cores 0, 1, 2, 3, 4 and 5 are Silver cores running at 1GHz and CPU cores 6, 7 are Gold cores running at 1.7GHz.

As per section 4.2. of this confluence page, Maximum DMIPS per Silver core @1.0GHz = 2,660 DMIPS/MHz.

Total DMIPS of silver CPU cores 3, 4, 5 = 3 x 2660 = 7980 DMIPS/MHz

Hence, DMIPS consumed by iperf = 34% of (DMIPS of CPU cores - 03, 04, 05)

= 0.34 x 7980 = 2713 DMIPS/MHz

Result:

DMIPS consumed by iperf and its related receive interrupts be 2713 DMIPS/MHz (approx.)

5. PHY register MDIO read/write through sysfs

1. Command to read PHY register: -

Ex 1:

cd /sys/devices/platform/soc/20000.qcom,ethernet

echo cl45r 1 0x0834 > stmmac

cat stmmac

O/p - 0xc000

Ex 2:

echo cl45r 1 0x0002 > stmmac

cat stmmac

O/p - 0x2b (PHY ID 1)

Ex 3: -

echo cl45r 1 0x0003 > stmmac

cat stmmac

O/p - 0x0b21 (PHY ID 2)

2. **Command to write to PHY register: -**

6. Few QNX Hypervisor concepts

6.1. Meaning of "pass loc mem"

Configuration varianble: pass

Configure physical devices or data blob types passed through from the host to a guest.

Syntax:

```
[blob_type] pass [loc options] [intr guest_intr[=vector_number]]

loc location_spec[,length][,(+|-|r|w|x|c|e|m|d|n|s|t)*]|[=host_address]
Example: -
```

```
pass loc mem:0x00020000,0x010000,rw=0x00020000

The above creates a memory map of size 0x10000 or 64Kbytes located in the host-physical memory at address
0x00020000, accessible by the guest at address 0x00020000 in guest-physical memory.
This region is read/write capable due to rw notation.

For more information, watch below URL:
https://www.qnx.com/developers/docs/7.1/index.html#com.qnx.doc.hypervisor.user/topic/vm/pass.html
```

Note: -

1. Switching between Android and QNX console: -

TML_Gen3_Procedure.txt

References

https://stackoverflow.com/questions/1221555/retrieve-cpu-usage-and-memory-usage-of-a-single-process-on-linux

Ethernet Hardware Pass Through from HQX

03 Performance KPIs

How to calculate Max. DMIPS of SoC