组会报告

徐益

2018 年 8 月 23 日

1 工作内容

1. 学习 High-Throughput Multi-Core LDPC Decoders Based on x86 Processor。
2. 学习相关代码。

2 论文对 layered-ms 算法的改进

2.1 原数据结构

```
int t_deg_cn[n-k] = { 4, 3, 4, 3, 4 };                              1
                                                                    2
unsigned short indicies[ ] = {                                      3
        0, 1, 2, 3, // VN nodes for C0                              4
        3, 4, 5,    // VN nodes for C1                              5
        1, 4, 6, 7, // VN nodes for C2                              6
        0, 3, 6,    // VN nodes for C3                              7
        0, 2, 3, 6  // VN nodes for C4                              8
};                                                                  9
                                                                   10
// decoder description (init + iteration loop)                     11
for(n=0; n<C; n++)                                                 12
  unsigned char deg_cn = t_deg_cn[n];                              13
  for(i = 0; i < deg_cn; i++){ /* COMPUTE CN value */ }            14
  for(i = 0; i < deg_cn; i++){ /* UPDATE MSG&VN values */ }        15
}                                                                  16
// end of decoding process description (hard decision)             17
```

图 1: Naive decoder kernel description using constant arrays

空间占用情况:

$$\Delta = 4 \times n + 4 \times m + 2 \times m + (n - k). \tag{1}$$

2.2 msg 类型优化

从 float 变成 int8_t
空间占用情况:

$$\Delta = n + m + 2 \times m + (n - k). \tag{2}$$

2.3 基于交织的并行计算

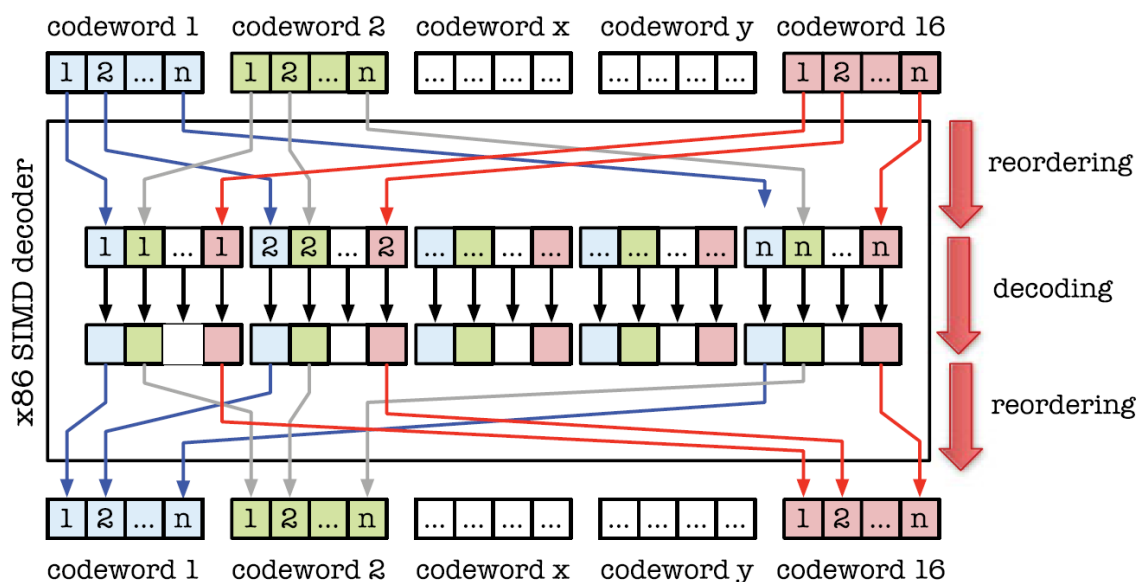图 2: Data interleaving and desinterleaving processes

空间占用情况：

$$\Delta = q \times n + q \times m + 2 \times m + (n - k). \tag{3}$$

2.4 重新排布校验矩阵

$$
\begin{array}{c}
\begin{array}{cccccccc}
 & V_0 & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7
\end{array}\\
\begin{array}{c}
C_0[\delta_0=4]\\
C_1[\delta_1=3]\\
C_2[\delta_2=4]\\
C_3[\delta_3=3]\\
C_4[\delta_4=4]
\end{array}
\left(\begin{array}{cccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0\\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 1\\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0\\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 0
\end{array}\right)
\end{array}
\Rightarrow
\begin{array}{c}
\begin{array}{cccccccc}
 & V_0 & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7
\end{array}\\
\begin{array}{c}
C_0[\delta_0=4]\\
C_2[\delta_1=4]\\
C_4[\delta_2=4]\\
C_1[\delta_3=3]\\
C_3[\delta_4=3]
\end{array}
\left(\begin{array}{cccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 1\\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0\\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0
\end{array}\right)
\end{array}
\Rightarrow
\begin{array}{c}
\begin{array}{cccccccc}
 & V_0 & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7
\end{array}\\
\begin{array}{c}
C_0[\delta_0=4]\\
C_4[\delta_4=4]\\
C_2[\delta_2=4]\\
C_1[\delta_1=3]\\
C_3[\delta_3=3]
\end{array}
\left(\begin{array}{cccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0\\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 0\\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 1\\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0\\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0
\end{array}\right)
\end{array}
$$

图 3: CN Computation Reordering

空间占用情况：

$$\Delta = q \times n + q \times m + 2 \times m. \tag{4}$$

2.5 其他

1. 预先计算好 En 地址;
2. 使用最新的指令集;
3. 多核计算。

3 代码学习

Algorithm 1. Horizontal TDMP Min-Sum algorithm

1: **Kernel 1**: Initialization
2: **for all** $m \in C, n \in \Psi(m)$ **do**
3:     $L_{mn}^{(0)} = 0$
4: **end for**
5:   $\triangleright$ *Process iter_max decoding iterations*
6: **for all** $t = 1 \rightarrow (iter\_max)$ **do**
7:     **Kernel 2**: For each check node in the code
8:     **for all** $m \in C$ **do**
9:         $\triangleright$ *Compute L_{nm} message*
10:         **for all** $n \in \Psi(m)$ **do**
11:             $L_{nm}^{(t)} = E_n - L_{mn}^{(t-1)}$
12:         **end for**
13:         $\triangleright$ *Compute L_{mn} message*
14:         **for all** $n \in \Psi(m)$ **do**
15:           $sign(L_{mn}^t) = \left[\prod_{(n' \in \Psi(m)/n)} sign(L_{n'm}^{(t)}) \right]$
16:           $|L_{mn}^t| = \left[\min_{(n' \in \Psi(m)/n)} |L_{n'm}^{(t)}| \right]$
17:         **end for**
18:         $\triangleright$ *Immediately update E_n*
19:         **for all** $n \in \Psi(m)$ **do**
20:             $E_n = L_{nm}^t + L_{mn}^t$
21:         **end for**
22:     **end for**
23: **end for**
24: **Kernel 3**: Hard decision
25: **for all** $n \in V$ **do**
26:     $\hat{c}_n = \begin{cases} 0 & \text{if } E_n \leq 0 \\ 1 & \text{if } E_n > 0 \end{cases}$
27: **end for**

4　存在问题


```
ipo: warning #11021: unresolved _ZNKSt5ctypeIcE13_M_widen_initEv
        Referenced in /tmp/ipo_icc7h0DvI.o
ipo: warning #11021: unresolved _Znwm
        Referenced in /tmp/ipo_icc7h0DvI.o
ipo: warning #11021: unresolved _ZNKSt7__cxx1112basic_stringIcSt11char_traitsIcE
SaIcEE7compareEPKc
        Referenced in /tmp/ipo_icc7h0DvI.o
ipo: warning #11021: unresolved _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcES
aIcEE9_M_createERmm
        Referenced in /tmp/ipo_icc7h0DvI.o
ipo: warning #11021: unresolved _ZSt19__throw_logic_errorPKc
        Referenced in /tmp/ipo_icc7h0DvI.o
ipo: warning #11021: unresolved _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcES
aIcEE10_M_replaceEmmPKcm
        Referenced in /tmp/ipo_icc7h0DvI.o
ipo: warning #11021: unresolved __cxa_call_unexpected
        Referenced in /tmp/ipo_icc7h0DvI.o
ipo: warning #11021: unresolved _ZSt4cout
        Referenced in /tmp/ipo_icc7h0DvI.o
ld: 找不到 -lboost_system-mt
ld: 找不到 -lboost_timer-mt
Makefile:44: recipe for target 'main.icc' failed
make: *** [main.icc] Error 1
sherlockhsu@lab:~/Github/Fast_LDPC_decoder_for_x86/bin$
```

5　下阶段计划

1. 使程序正常运行；
2. 尝试与原仿真程序结合。