Abstract

Continuous improvement in silicon process technologies has made possible the integration of hundreds of cores on a single chip. However, power and heat have become dominant constraints in designing these massive multicore chips causing issues with reliability, timing variations and reduced lifetime of the chips. Dynamic Thermal Management (DTM) is a solution to avoid high temperatures on the die. Typical DTM schemes only address core level thermal issues. However, the Network-on-chip (NoC) paradigm, which has emerged as an enabling methodology for integrating hundreds to thousands of cores on the same die can contribute significantly to the thermal issues. Moreover, the typical DTM is triggered reactively based on temperature measurements from on-chip thermal sensor requiring long reaction times whereas predictive DTM method estimates future temperature in advance, eliminating the chance of temperature overshoot. Artificial Neural Networks (ANNs) have been used in various domains for modeling and prediction with high accuracy due to its ability to learn and adapt. This thesis concentrates on designing an ANN prediction engine to predict the thermal profile of the cores and Network-on-Chip elements of the chip. This thermal profile of the chip is then used by the predictive DTM that combines both core level and network level DTM techniques. On-chip wireless interconnect which is recently envisioned to enable energy-efficient data exchange between cores in a multicore environment, will be used to provide a broadcast-capable medium to efficiently distribute thermal control messages to trigger and manage the DTM schemes.

Artificial Neural Network to Predict Temperature profile in a Multi-core chip The proactive Dynamic Thermal Management (DTM) mechanism in this work, is triggered using Artificial Neural Network (ANN) based temperature predictor. Future temperature is estimated from the performance counters or utilization metrics in advance. This prediction based DTM eliminating the chance of transient overshoot of temperature. As a result, the impact on performance is minimal for these types of predictive thermal estimator [3]. As these thermal estimation schemes need to be implemented on-chip, it is important for these mechanisms to have low computational and area overheads. ANN based prediction mechanism has been shown to perform with high accuracy in various application areas [5]. This work designs a hardware-based ANN thermal predictor that is trained and apply to predict the temperature at any given time based on the utilization of the chip components. An ANN consists of two elements: first element adds products of input and weights coefficients. The second element is a neuron activation function which is a nonlinear function, as explained in section 1.3. The system dependency between input and output can be modeled using the ANN, on training the ANN on a train data set. The training data consists of inputs i.e. utilization of the chip components and their corresponding output i.e. temperature increases due to the utilization of the element. ANN are Multi-Input Multi-Output model that demand hardware overhead. To reduce the hardware overhead and reuse the available resources, the neurons are realized as parallel multiply-accumulate operation (MAC) units. 16 3.1 Creating the training dataset The desire to have an ANN to model the dependency between utilization of the core and NoC level components with the temperature various components of the chip (cores/switches/links), creates the necessity to generate a training dataset. The utilization of the cores are represented by the percentage utilization of the processors while the utilization of the switches are measured as the ratio of actual buffer occupancy to that of the maximum. The utilization of the links are measures at the attached switches as the ratio of the actual rate of flits transferred over the link to the maximum capacity of the link. All the utilizations are expressed as a percentage. To gather the training data, first random initializations of utilizations of core level and NoC level component is done on a cycle accurate in-house multi-core NoC simulator. The simulator is allowed to run for 3000 cycle with the initialized conditions. At the end of 3000 cycles simulator outputs the temperature change in each cycle. Similarly several random utilization conditions are initialized to the simulator, then the corresponding output temperatures are recorded as the training data. The total training data that was gathered is 3000 x 250 samples. This would help for the prediction of temperature in that particular help in the operating range of the system. 3.2 Design of the ANN The inputs to the ANN are the utilization of all the network element. The output of Base ANN is the temperatures of the cores, switches and links. The Base ANN output is compared to the threshold producing a single bit output. This output signifies 17 if the element has crossed the threshold or not. The trained ANN is a concatenation of three subdivided ANN streams. The subdivided ANN streams are Core streams, Link stream and Switch stream. The inputs to all the streams remain the same, i.e. the utilization of all the network elements and time to be predicted This structure of ANN results in better accuracy and decreases the number of connections between hidden layer and output layer i.e. the fully connected neurons exists only between the hidden neurons of cores and output neurons of the cores. The reduced number of connections between the hidden neurons and the output neurons decreases the latency of prediction. . The nntool box of Matlab is used to train Fig. 3: Trained ANN structure. 18 to neural network. The above Fig.3 shows the trained ANN structure. The number of hidden layer neurons used for the Cores stream is 250 whereas for the Switches and Links stream are 50 and 100 respectively. These hidden layer neurons were selected by a number of trials, resulting for the best accuracy and low latency. There were a number of experiments done using different kinds of activations function but the best accuracy was produced by using the sigmoid activation function for hidden layer and linear activation function for the output layer. To improve the latency of calculation, the number of neurons were decreased from 700 to 400. It was observed that there is negligible loss on accuracy for change in the number of neurons. 3.3 ANN hardware To have an efficient ANN realization with the use of available resource in a core, the neurons are realized as multiplier and accumulator (MAC) units and work in parallel. Fig.4 shows the designed ANN hardware. The computation to predict the temperature, starts no sooner the utilization information packets are received at the ANN core in the pipelined fashion. This utilization values are represented as 8 bit numbers. We propose to use wireless interconnections for the transfer of information to and from the ANN to the different components of the chip. The components packetize their current utilization values and send it to the ANN via the nearest NoC switch. The only wireless switch with the token to pass network utilization can send the information. There are 240 REG arrays each of size 8 bit. 19 The number of neuron operations that are occurring in parallel are equal to the number of available MACs. As the number of MAC units that work in parallel are less than the number of inputs, we use register arrays to store the inputs. Inputs are being multiplied with the corresponding weight values that are fetched from the weight memory. The. FSM control unit synchronizes the received inputs, weights and directing these values to the right MAC unit. MAC outputs are stored in the cache. When all the connections to a particular neuron are complete then the output is passed through the activation function via buffer. The output of the LUT is feedback to the input coordination unit and REG array, only if the computation was done between first two layers (input layer and hidden layer). The process is repeated for the hidden layer and Fig. 4: ANN hardware 20 the output layer. For the first iteration the predicted increase in temperature is added to the initial temperature of the chip and then accumulated. From the second iteration onwards predicted temperature is added to the accumulated result. The accumulator result is passed through a comparator, where it is compared with the threshold. The comparator outputs a single binary bit. This bit signifies that, the corresponding network element is in the 'ON' state and the zero bit signifies the ‘OFF’ state of the element. The comparator output is broadcasted wirelessly, which is used by network elements for efficient Distance Vector Routing (DVR) and Task Rerouting (TR). The ANN predicts the future temperature for a given utilization every 100,000 cycles. Hence, the bandwidth demand of these DTM related packets on the wireless interconnection is 0.054Gbps. This being a small fraction of the wireless bandwidth of 16Gbps does not significantly affect the performance. For efficient DVR and TR to be performed that results in a chip temperature less than the threshold, the 240 status bits are to be transmitted within 1,000,000 clock cycles