Proceedings of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, May 27-29, 2006 (pp82-87)

Hardware/Software Co-Design Using Bayesian Belief Networks

Ahmed Sameh, and Waseem Raslan
Department of Computer Science
The American University in Cairo
P.O.Box 2511, Cairo, Egypt

Abstract

HW/SW techniques make it possible for the system designers to validate their design, assign modules to be implemented in either hardware or software in the early stages of the system design life cycle. In addition, those techniques provide powerful mechanism for continuous system validation until the final product is done. Partitioning the system into either hardware or software, in the system early stages, is vital decision that has to be done iteratively and accurately. Many techniques have been proposed for HW/SW partitioning: conventional circuit partitioning techniques, simulated annealing, expert systems, and even genetic algorithm techniques. The partitioning problem has been proved to be and NP-Hard problem, thus AI, ANN and GA techniques can find a rich playground to apply their techniques. This paper presents a novel approach to use Bayesian Belief Networks as the tool that does the partitioning decision when provided by simulation parameters that measure certain characteristics in the design.

1. Introduction

Hardware/Software co-design can be viewed as the marriage between the hardware and software life cycles to reduce the amount of integration problems used to appear when integrating SW and HW in the unified system (figure 1). It's considered the new optimum development environment for embedded systems that have penetrated approximately every modern appliance nowadays. It's also very helpful to co-design Application Specific Instruction Set Processors ISA. In addition to being helpful to design re-configurable systems that can be personalized after being manufactured. Its effectiveness affects the system turn-over time from design to consumer.

Many aspects are involved in HW/SW design: using unified language for both of HW and SW modeling, simulation and synthesis. Verification of both the system components is essential to ensure that system testing is being performed in at any stage in the HW/SW co-design life cycle. HW/SW partitioning is another important aspect when talking about HW/SW co-design.

Partitioning is the process of assigning each of system components to implemented in either HW or SW. Objectives of any partitioning techniques are to maximize speed, latency requirements, and minimize size, power and cost. Being part of an iterative process, the partitioning technique must have a very good performance. Unfortunately, the problem has been proved to be an NP-hard problem. This process is vital for the HW/SW co-design since it affects the overall system cost and performance. Granularity of the system components, course versus fine granularity affect the partitioning technique efficiency. Re-targetability, or the ability of the system to be independent of the microcontroller or microprocessor that will be used to run the SW part of the system, is another issue for the partitioning technique.

2. A BBN for Co-Design Partition

The shareware version of Hugin software systems (HuginLite) is used to verify the idea of using the BBN in HW/SW partitioning. The block diagram design of a programmable thermostat has been reduced into only five main components: Program Interaction, Control Temperature, User Interaction, Time Keeping and Get Current Temperature. Using the StateMate HW/SW environment the design block diagram has been transferred to its BBN representation as shown in figure (2). Characteristic probing statements have been added to the design models to extract useful information about the design like: frequency of execution, bandwidth and complexity measures. Utilizing those simulation measures the link matrix between any two BBN nodes are evaluated to give the link matrices shown in the following figure (2).

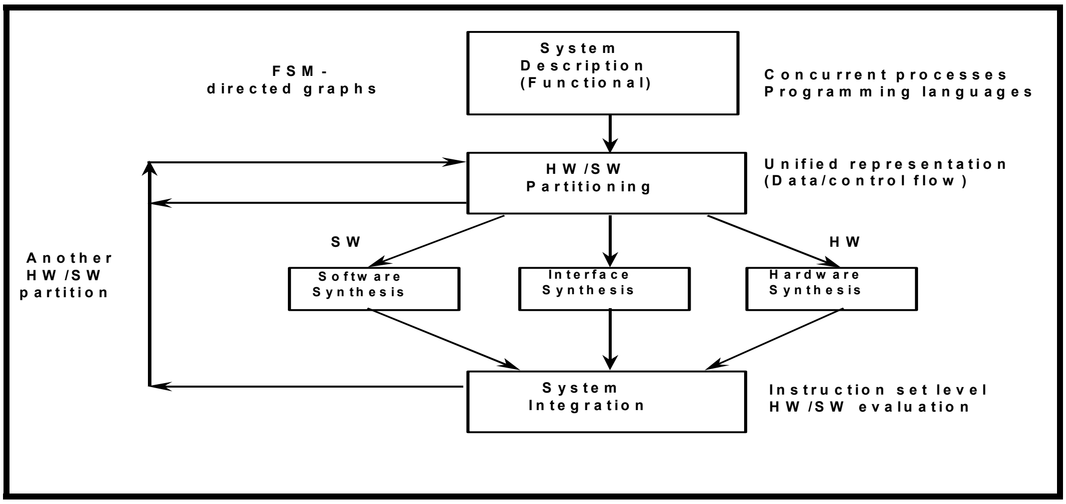


Figure 1 HW/SW Partitioning in co-design flow

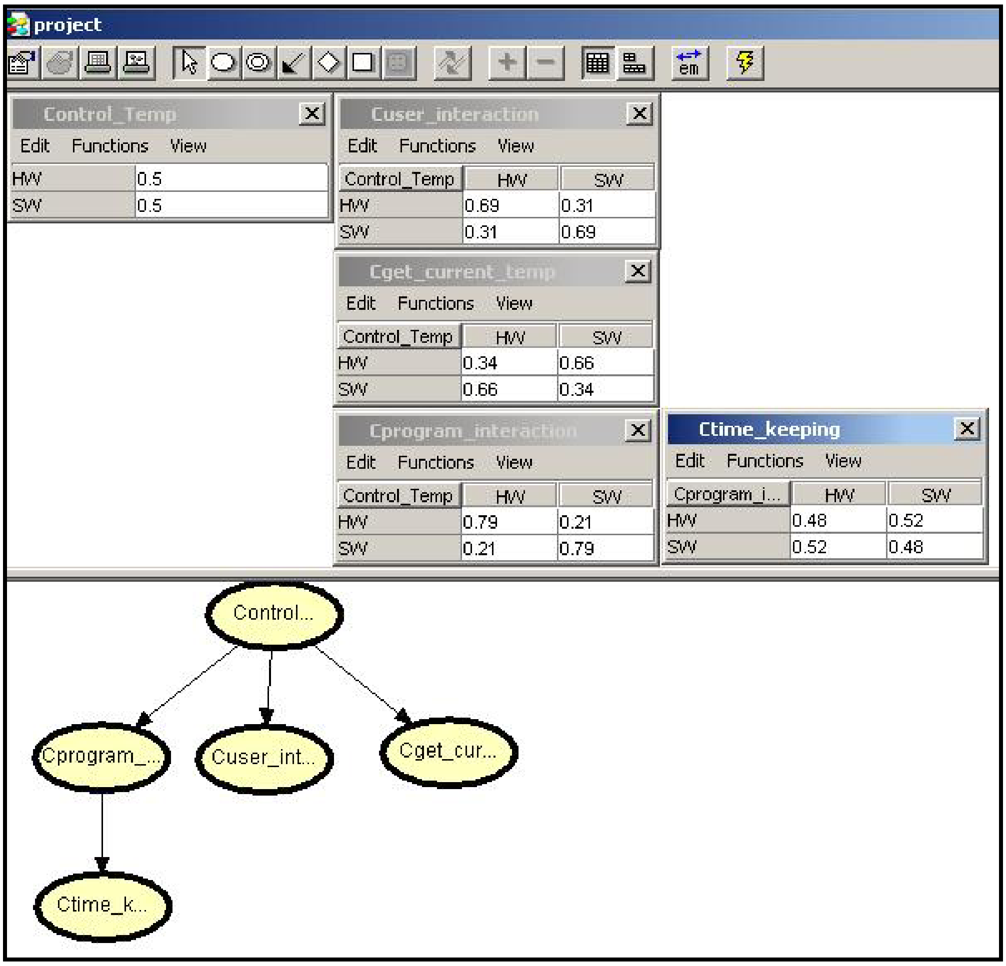


Figure 2 BBN for the programmable thermostat with the Link Matrices specified

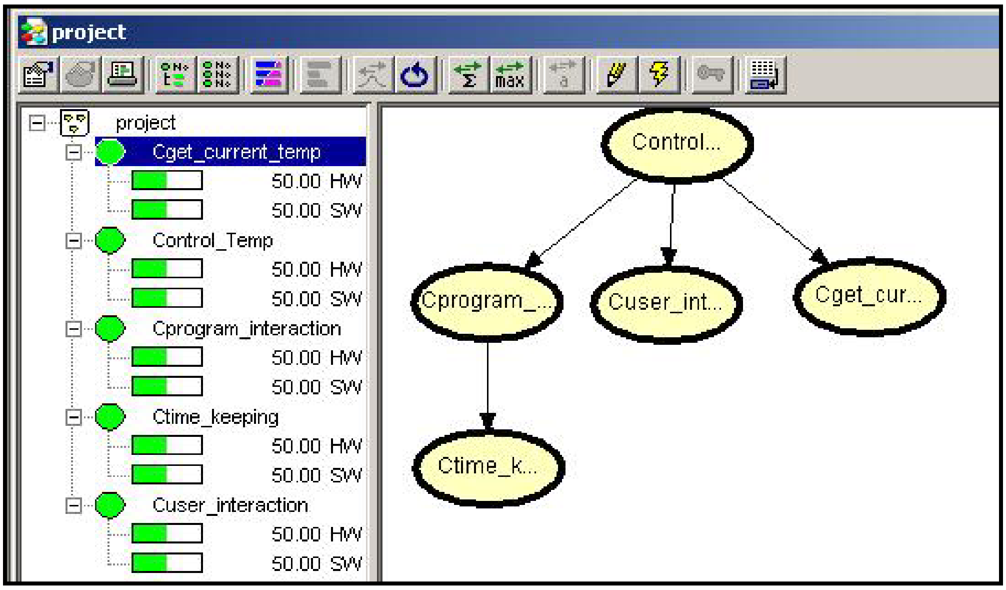


Figure 3 Initial guesses for HW/SW probabilities are equal

The HW/SW balance is 50%-50% at the beginning as it can be seen in figure (3). Hence evidences are introduced to various design components the Hugin tool carries the task of propagating the every new evidence (entered as likelihood of implementing each model either in HW or SW) through the whole BBN giving new HW/SW percentages for each node/component.

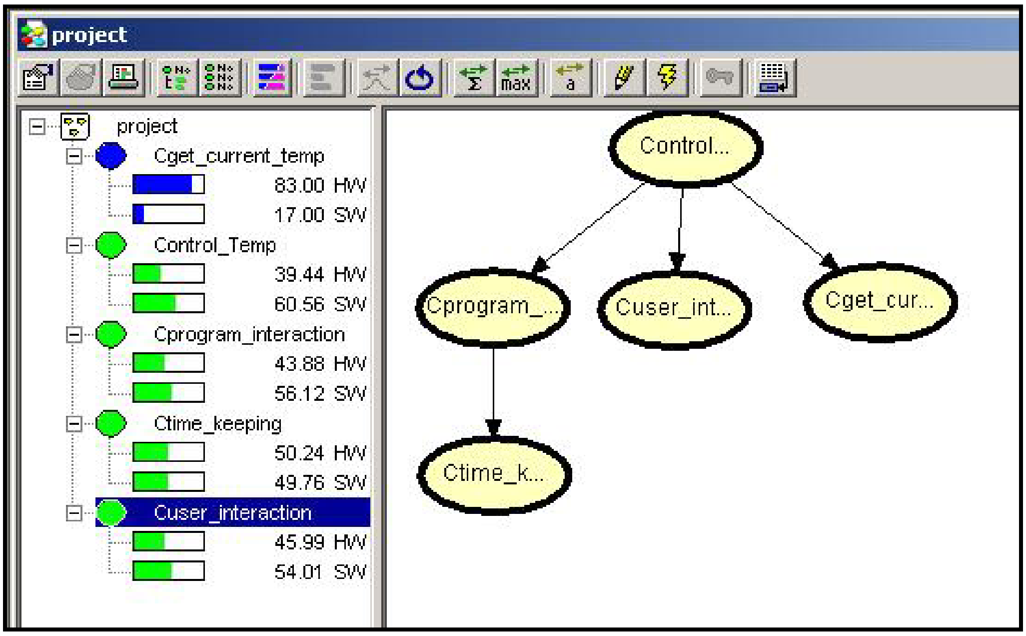


Figure 4 BBN After applying evidence found on Get-Current-Temp component

Figure (4) shows the new HW/SW likelihood after applying evidence found on the Get_Current_Temp component to have 83%-17% HW-SW implementation ratio. As it can be seen from the link matrix between Control_Temp and Get_Current_Temp components if the later is driven to the SW direction the former is driven to the HW direction by a 0.66/0.34 ratio and vice versa. Thus since Get_Current_Temp has been driven into HW, Control_Temp has been driven to the SW direction.

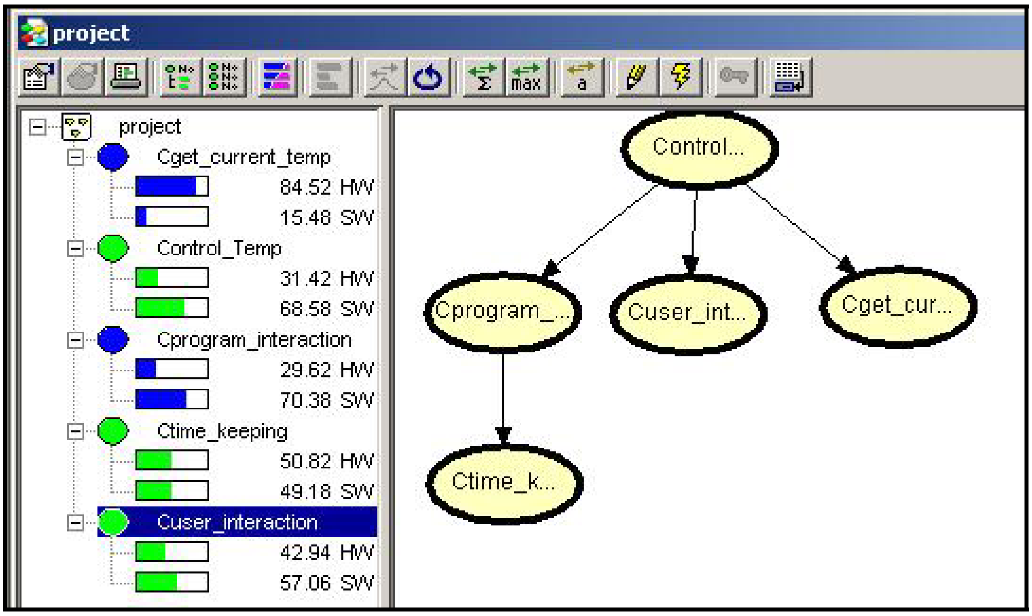


Figure 5 BBN After applying the second evidence found on the Program-Interaction component

Applying another evidence that the Program interaction component has a 35% likelihood to be implemented in HW, and propagating the evidence through the BBN the new HW/SW likelihood is presented in figure (5).

A threshold, 75% is suggested as a percentage that can be selected so that any percentage that exceeds it will be implemented as it indicates.

3. Extended BBN

Collapsing the set of design metrics into a single one using the parameters used in the previous figures is not very intuitive as it hides the true contribution of each metric on the design component and hides the relationship between dependent metrics. Figure 6 suggests separating the links between each two nodes into a number of links equal to the number of design metrics (complexity, bandwidth, and frequency of execution).

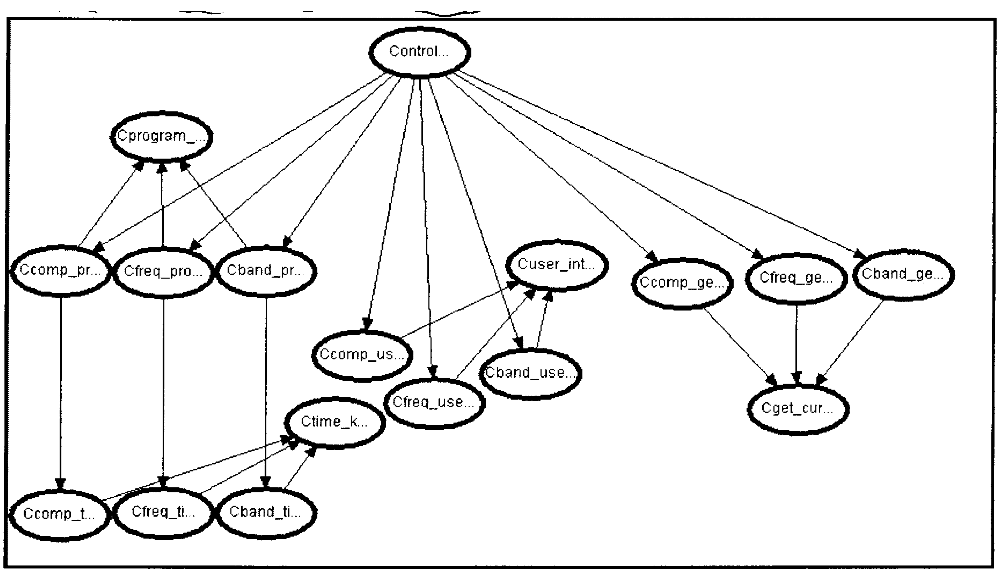
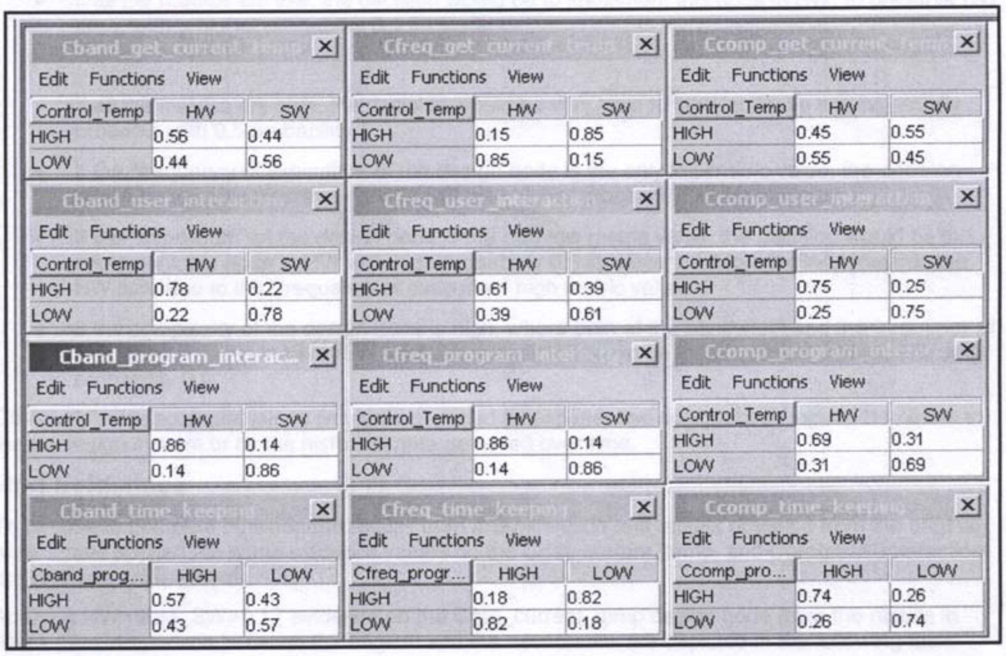


Figure 6 New Design for the BBN model

Calculating the link matrices between each of the metric nodes and its affecting node is straightforward and follows the same original equations. The new link matrices, after normalization, between the metrics nodes and their direct affecters appear in figure 7.



Control_Temp	HW	SW
HIGH	0.56	0.44
LOW	0.44	0.56

Control_Temp	HW	SW
HIGH	0.15	0.85
LOW	0.85	0.15

Control_Temp	HW	SW
HIGH	0.45	0.55
LOW	0.55	0.45

Control_Temp	HW	SW
HIGH	0.78	0.22
LOW	0.22	0.78

Control_Temp	HW	SW
HIGH	0.61	0.39
LOW	0.39	0.61

Control_Temp	HW	SW
HIGH	0.75	0.25
LOW	0.25	0.75

Cband_program_interac...

Control_Temp	HW	SW
HIGH	0.86	0.14
LOW	0.14	0.86

Control_Temp	HW	SW
HIGH	0.86	0.14
LOW	0.14	0.86

Control_Temp	HW	SW
HIGH	0.69	0.31
LOW	0.31	0.69

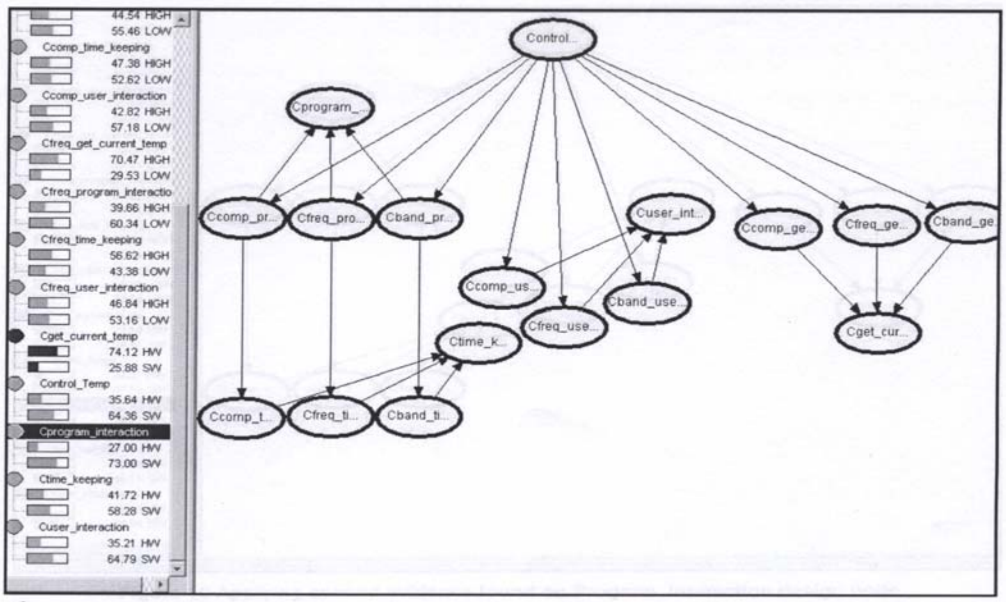
Cband_prog...	HIGH	LOW
HIGH	0.57	0.43
LOW	0.43	0.57

Cfreq_progr...	HIGH	LOW
HIGH	0.18	0.82
LOW	0.82	0.18

Ccomp_pro...	HIGH	LOW
HIGH	0.74	0.26
LOW	0.26	0.74

Figure 7 Link Matrices between metrics nodes and their direct affecters

The remaining uncalculated link matrices are those between the metrics nodes and the design nodes they directly affect. Here comes the designer's role and his past design experience to define the actual relationship between the different metrics and how they could affect the design decision. Figure 8, shows the results of applying HW=0.83, SW=0.17 on the Cget_current_temp design node as evidence. Figure 9, shows the results of applying HW=0.65, SW=0.35 on the Program_interaction design node as evidence



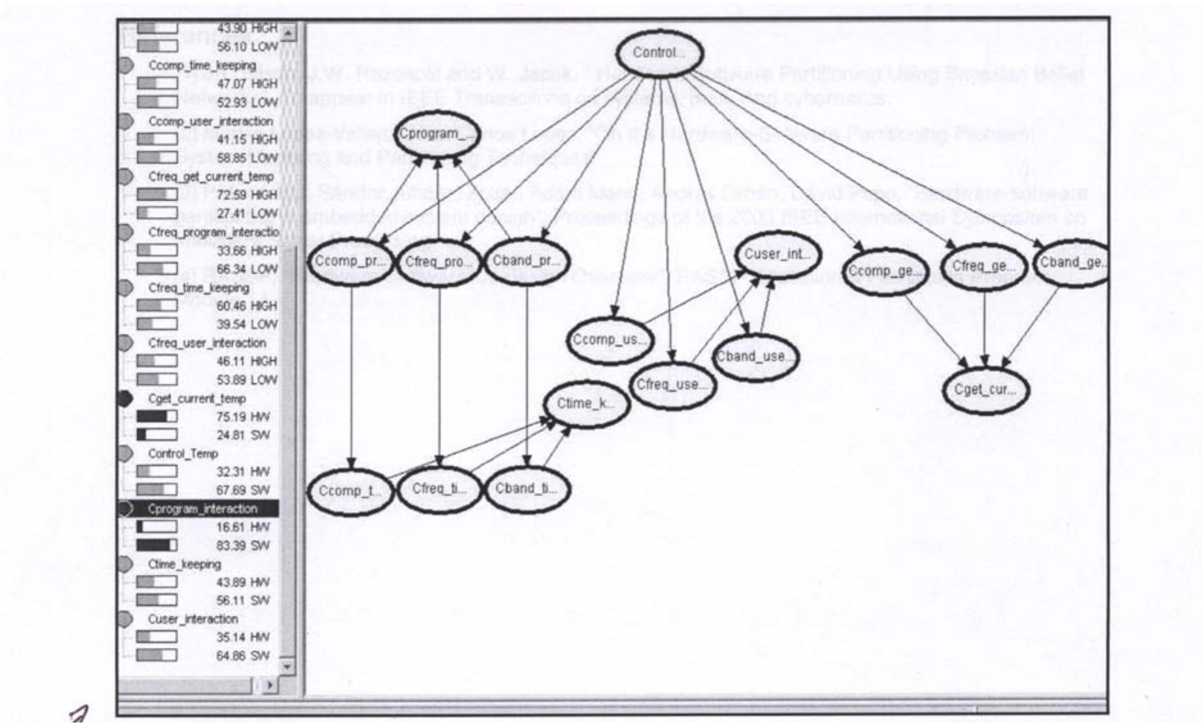


Figure 8&9: Applying Evidence on specific nodes

4. Conclusion

The HW/SW partitioning problem remains open for researcher efforts to obtain optimum results at reasonable performance. This paper presented a novel approach to utilize the Bayesian Belief Network engine as the tool to examine and decide on the design partitioning problem. Determining the link matrix values between different nodes is the core of this technique, in this paper a methodology to calculate the link matrices using the cost, frequency of execution and complexity have been provided and validated through experiments. An extended BBN model has been suggested that preserves the different metrics contributions on the HW/SW decision. The new results are acute to the original results, yet there is a need for an accurate, designer history based mechanism to define the link matrix between the metrics nodes and the design nodes. The results are sound but extra work needed to validate the new methodology and compare it against the other partitioning techniques. Integrating the BBN methodology, through using the available BBN API in the design environment would have given more flexibility and ability to compare against other techniques.

5. References

[1] J.T. Olson, J.W. Rozenblit and W. Jacak, " Hardware/Software Partitioning Using Bayesian Belief Networks". To appear in IEEE Transactions on systems, man. And cybernetics.

[2] Marisa Lopez-Vallejo, Juan Carlos Lopez, "On the Hardware-Software Partitioning Problem: System Modeling and Partitioning Techniques"

[3] Péter Arató, Sándor Juhász, Zoltán Ádám Mann, András Orbán, Dávid Papp, "Hardware-software partitioning in embedded system design", Proceedings of the 2003 IEEE International Symposium on Intelligent Signal Processing.

[4] RASSP, "Hardware/Software Co-design Overview", RASSP Education & Facilitation Program, Module 14